



Network of interacting neurons with random synaptic weights

Paolo Grazieschi, Marta Leocata, Cyrille Mascart, Julien Chevallier, François Delarue, Etienne Tanré

► To cite this version:

Paolo Grazieschi, Marta Leocata, Cyrille Mascart, Julien Chevallier, François Delarue, et al.. Network of interacting neurons with random synaptic weights. ESAIM: Proceedings and Surveys, 2019, CEMRACS 2017 - Numerical methods for stochastic models: control, uncertainty quantification, mean-field, 65, pp.445-475. 10.1051/proc/201965445 . hal-01928990

HAL Id: hal-01928990

<https://inria.hal.science/hal-01928990>

Submitted on 20 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NETWORK OF INTERACTING NEURONS WITH RANDOM SYNAPTIC WEIGHTS

PAOLO GRAZIESCHI, MARTA LEOCATA, CYRILLE MASCART, JULIEN CHEVALLIER, FRANCOIS DELARUE,
AND ETIENNE TANRE

ABSTRACT. Since the pioneering works of Lapicque [17] and of Hodgkin and Huxley [16], several types of models have been addressed to describe the evolution in time of the potential of the membrane of a neuron. In this note, we investigate a connected version of N neurons obeying the leaky integrate and fire model, previously introduced in [1, 2, 3, 7, 6, 15, 18, 19, 22]. As a main feature, neurons interact with one another in a mean field instantaneous way. Due to the instantaneity of the interactions, singularities may emerge in a finite time. For instance, the solution of the corresponding Fokker-Planck equation describing the collective behavior of the potentials of the neurons in the limit $N \rightarrow \infty$ may degenerate and cease to exist in any standard sense after a finite time. Here we focus out on a variant of this model when the interactions between the neurons are also subjected to random synaptic weights. As a typical instance, we address the case when the connection graph is the realization of an Erdős-Renyi graph. After a brief introduction of the model, we collect several theoretical results on the behavior of the solution. In a last step, we provide an algorithm for simulating a network of this type with a possibly large value of N .

1. INTRODUCTION

1.1. From a model for one neuron to a model for N neurons. One of the first models for neurons was introduced by Louis Lapicque in 1907, see [17]. It is called Integrate and Fire (IF) model. The membrane potential $(X(t))_t$ of a neuron evolves in time according to the simple electrical equation

$$I(t) = C \frac{dX(t)}{dt},$$

where $(I(t))_t$ is an input current that goes through the membrane. The above equation holds true up until the neuron emits a *spike*. We use the generic notation τ_F for the *spiking time*. From the mathematical point of view, τ_F is regarded as the first time when the membrane potential reaches a certain hard threshold value X_F , the latter being referred to as a *firing value*. Equivalently, the time τ_F is nothing but the first *hitting time* of X_F . As just mentioned, it must be seen as a *firing time* at which the neuron emits a *spike*. After τ_F , the dynamics are continued in the following way: At time τ_F , the potential is reset to a reset value X_R and the dynamics are refreshed from X_R . This model serves as the basis for a large variety of other neural network models developed in order to model more accurately certain behaviors of neurons and neural networks, such as memory, leaking, etc. For instance, a common feature is to write the *sub-threshold dynamics* of the membrane potential in the form of a stochastic (instead of ordinary) differential equation:

$$dX_t = b(X_t)dt + I_t dt + dW_t,$$

for $t < \tau_F$, where $(I_t)_t$ describes the mean trend of the input current (we here use the probabilistic convention: the time parameter is put in index) and $(\frac{d}{dt}W_t)_t$ is a white noise accounting for fluctuations in the input. A typical choice for the coefficient b is $b(x) = -\lambda(x - a)$, in which case λ reads for some leakage parameter. This model is commonly referred to as a stochastic LIF model.

In this article, we address a network of N connected neurons that evolve according to a variant of the above stochastic LIF model. We describe the discrete system of neurons through the evolution in time of the membrane potentials of each of the neurons: The membrane potential of neuron i (for $i = 1, \dots, N$) is described by a process $X^i = (X_t^i)_{t \geq 0}$. The *sub-threshold* dynamics of each X^i are of the general form:

$$dX_t^i = b(X_t^i)dt + dI_t^{i, \text{network}} + dW_t^i, \quad t \geq 0, \quad i = 1, \dots, N,$$

with initial conditions $X_0^i < X_F$. Here $(X_0^i)_{i=1, \dots, N}$ is a family of independent and identically distributed random variables and $(W^i)_{i=1, \dots, N}$ is a family of independent Brownian motions, the initial conditions and the noises being independent. The term $dI_t^{i, \text{network}}$ models the current that neuron i receives at time t from the other neurons. As above, the value X_F is a firing threshold, which is assumed to be common to all the neurons: Whenever a neuron reaches the threshold value X_F , its potential is reset to X_R , X_R being the same for all the neurons.

For sure, the term $dI_t^{i,\text{network}}$ is of great importance in the description of the model. Inspired by [18, 22], we choose $dI_t^{i,\text{network}}$ in the form

$$dI_t^{i,\text{network}} = \sum_{j=1}^N \sum_{k \geq 1} J_N^{j \rightarrow i} d\delta_0(t - \tau_k^j),$$

where δ_0 is the Dirac delta measure in 0, τ_k^j (for some $j = 1, \dots, N$ and $k \in \mathbb{N}$) is the k -th time at which the potential of neuron j reaches the threshold value X_F , and $J_N^{j \rightarrow i}$ is a *synaptic weight* that prescribes the influence of neuron j onto neuron i . In case when the synaptic weight is positive, the interaction from j to i is excitatory; otherwise, it is inhibitory. It is worth mentioning that, with the above prescription, $(dI_t^{i,\text{network}})_t$ is not exactly a current since the measure $dI_t^{i,\text{network}}$ is singular. Differently, the term $dI_t^{i,\text{network}}$ account for impulses received by neuron i from the network at time t . These impulses account for the instantaneity of the interactions: each time neuron j spikes, neuron i feels a pulse of intensity $J_N^{j \rightarrow i}$.

The main purpose of this note is to address the case when the synaptic weights are of the form

$$J_N^{j \rightarrow i} = \frac{\beta}{N} \alpha^{i,j},$$

where β is a scaling parameter that calibrates the global connectivity of the network and $\alpha^{i,j}$ represents some variability in the synaptic weight between neurons i and j . The factor $1/N$ is typical of a *mean field* model. The main feature is that we allow the weights $(\alpha^{i,j})_{i,j=1,\dots,N}$ to be random, in which case the tuples $(X_0^i)_{i=1,\dots,N}$, $((W_t^i)_{t \geq 0})_{i=1,\dots,N}$ and $(\alpha^{i,j})_{i,j=1,\dots,N}$ are required to be independent.

The idea below is to study the behavior of the system, when N is large, considering different kind of randomness for $\alpha^{i,j}$.

1.2. Mean field limit. The time integrated version of the dynamics may be put in the more accessible form:

$$(1) \quad \begin{cases} X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \sum_{j=1}^N \sum_{k \geq 1} J_N^{j \rightarrow i} \mathbb{1}_{\{\tau_k^j \leq t\}} + W_t^i, & \text{if } X_{t-}^i < X_F, \\ X_t^i = X_R, & \text{if } X_{t-}^i = X_F. \end{cases}$$

The case $J_N^{j \rightarrow i} = \alpha/N$, for a deterministic factor α , has been already addressed in several papers, among which [1, 2, 3, 7, 6, 18, 22]. In all these references, authors have paid much effort to discuss the asymptotic regime $N \rightarrow \infty$.

Because of the normalization factor α/N in the interaction term and by independence of the various sources of noise, we may indeed expect from standard results on large particle system with mean field interaction, see for instance [24], that neurons become independent asymptotically, each of them solving in the limit a nonlinear stochastic differential equation with interaction with the common theoretical distribution of the network. In the limiting network, the membrane potential $(X_t)_t$ of a *representative neuron* should evolve according to the equation:

$$(2) \quad \begin{cases} X_t = X_0 + \int_0^t b(X_s) ds + \alpha \mathbb{E}[\sum_{k \geq 1} \mathbb{1}_{\{\tau_k \leq t\}}] + W_t, & \text{if } X_{t-} < X_F, \\ X_t = X_R, & \text{if } X_{t-} = X_F, \end{cases}$$

where $W = (W_t)_t$ is the proper noise to the representative neuron and $(\tau_k)_k$ stands for its own *spike train*. Passage from (1) to (2) is usually referred to as *propagation of chaos* in the literature. While it may be simpler to justify for systems with regular interactions, it turns out to be a much more challenging problem in the current framework because of the instantaneity of the interactions between the neurons.

In fact, even equation (2) itself –which should be called a McKean Vlasov equation– is a difficult object to study, especially in the excitatory regime $\alpha > 0$. In order to address its well-posedness, two approaches are conceivable, a PDE one and a probabilistic one. The PDE approach is based upon the description, through the so-called Fokker-Planck equation, of the evolution of the marginal law of the potential. We refer to [1, 2, 3]. The probabilistic route is to analyse directly equation (2), see [7, 6] and the more recent contributions [15, 19]. In fact, both approaches face the same difficulty: Whenever the interaction parameter α is too large, the system may **blow up** in some sense. Heuristically, a blow-up occurs when a large proportion of neurons in the finite network spike at the same time; in the limit regime, a blow up occurs when the time derivative of the mean field term in (2) is infinite. It is proven in [1, 6] that a blow up appears if the mass of the initial distribution is too concentrated near the firing threshold X_F and, conversely, that no blow up appears if the initial mass is sufficiently far away from X_F . While global existence of solutions with a blow up is addressed in [7] local uniqueness results are shown in [6, 15, 19]: In these latter references, successive improvements are obtained on the length of the interval on which uniqueness is known to hold.

Blow up not only matters for the limiting mean field equation. Going back to the particle system, we understand that the difference between the two values X_F and X_R is certainly important because it must influence, among others, the typical delay between two consecutive spikes. In particular, when α is large in

comparison with $X_F - X_R$, a neuron may spike at a high frequency because of the pulses received from the others. This stylized fact is at the core of our analysis. Still, the reader must be aware that, in biological models, an additional refractory period is usually added, during which a neuron is somehow locked after a spike. Here we shall not include such a refractory period; for sure, it would ask for further investigations.

Whilst the case $J_N^{j \rightarrow i} = \alpha/N$ has received much attention in the literature, the case of random synaptic weights has received limited attention. This is our precise objective to understand how randomness in the synaptic weights may impact the passage from (1) to (2) and how it may impact the emergence of a blow up in the system. In this regard, it is worth noticing that another form of inhomogeneous connection is addressed in [20]: Therein, a finite network of distinct infinite populations is studied; inhomogeneous weights are used to describe the connections between populations.

1.3. Model with random synaptic weights. Throughout the text, we will take $X_F = 1$ and $X_R = 0$. This permits to rewrite equation (1) for the network of $N \in \mathbb{N}$ neurons in the following more concise form:

$$(3) \quad X_t^i = X_0^i + \int_0^t b(X_s^i) ds + W_t^i + \sum_{j=1}^N J_N^{j \rightarrow i} M_t^j - M_t^i,$$

where the process M_t^i counts the number of times that the potential of neuron i has reached the threshold value $X_F = 1$; formally, we may write

$$M_t^i = \int_0^t \sum_{k \in \mathbb{N}} d\delta_0(s - \tau_k^i) = \sum_{k \in \mathbb{N}} \mathbb{1}_{[0,t]}(\tau_k^i),$$

with τ_k^i standing for the random time at which the process X^i reaches the threshold for the k -th time.

As already announced, we will discuss the behavior of the system for different types of variables $\{J_N^{j \rightarrow i}\}_{i,j=1,\dots,N}$. Particularly, we will address the case $J_N^{j \rightarrow i} = \beta \alpha^{i,j}/N$, with $\beta > 0$ constant and:

- (1) $\alpha^{i,j} \sim \mathcal{B}(p)$ i.i.d.;
- (2) $\alpha^{i,j}$ dependent with $\alpha^{i,j} = \alpha^i \alpha^j$ and $\alpha^i \sim \mathcal{B}(p)$ i.i.d.;
- (3) $\alpha^{i,j}$ dependent with $\alpha^{i,j} = p \alpha^i$ and $\alpha^i \sim \mathcal{B}(p)$ i.i.d.;
- (4) $\alpha^{i,j} \sim \mathcal{B}(p_N)$ i.i.d. with $\lim_{N \rightarrow \infty} p_N = 0$.

The note is organized as follows. We provide some theoretical results in Section 2. Section 3 is dedicated to the presentation of an algorithm for simulating (a variant of) the particle system with a large value of N . Throughout the text, all the Brownian motions are normalized, meaning that their volatility is 1.

2. MATHEMATICAL INQUIRIES

This section is dedicated to a theoretical analysis of (3), choosing $X_F = 1$ as firing threshold and $X_R = 0$ as firing potential. Throughout the section, we discuss (sometimes in a pretty informal way) the form of the limiting mean field equation together with the existence of a blow-up, as defined in the previous section.

2.1. Independent Random weights: $\alpha^{i,j} \sim \mathcal{B}(p)$. Let us consider the case where the connections are i.i.d. Bernoulli variables, with a parameter p that is independent of N , namely $\alpha^{i,j} \sim \mathcal{B}(p)$ (this is case (1) in Subsection 1.3). From a modeling point of view, this amounts to say that the neurons are connected through the realization of an Erdős-Renyi graph. The first purpose is to conjecture what is the limit equation, or equivalently the analogue of (2), for the network.

Observe first that, whenever $p = 1$, the synaptic weight takes the form $J^{j \rightarrow i} = \beta/N$, which exactly fits the case addressed in (2). By some heuristic calculation, we expect that, in the more general case $p \in (0, 1]$, the limiting mean field term manifests in the form:

$$p \cdot \beta \cdot \mathbb{E}[M_t].$$

Intuitively, the reason is that, as $N \rightarrow \infty$, not only the neurons should become asymptotically independent between them, but also they should become independent of the synaptic weights. Indeed a given neuron i feels less and less the values of each $\alpha^{i,j}$ as N grows up; this means that, applying a law of large numbers, the interaction term in (3) should get closer and closer to

$$\frac{\beta}{N} \mathbb{E} \left[\sum_{j=1}^N \alpha^{i,j} M_t^j \right] = \frac{\beta}{N} \mathbb{E} \left[\int_0^t \sum_{j=1}^N \sum_k \alpha^{i,j} d\delta(s - \tau_k^j) \right] = \frac{\beta}{N} \mathbb{E} \left[\sum_{j=1}^N \sum_k \alpha^{i,j} \mathbb{1}_{\{\tau_k^j \leq t\}} \right] \approx p \cdot \beta \cdot \mathbb{E}[M_t].$$

So, as a limit equation for the network, we expect the following (at least in the *sub-threshold regime*):

$$(4) \quad X_t = X_0 + \int_0^t b(X_s) ds + \beta \cdot p \cdot \mathbb{E}[M_t] + W_t - M_t,$$

with $M_t = \sum_{k \in \mathbb{N}} \mathbb{1}_{[0,t]}(\tau_k)$. In other words, the Bernoulli parameter should scale as a linear factor in the intensity of the interaction. In particular, if our derivation is correct, the limiting equation fits the framework considered in [1, 2, 3, 7, 6]. We will come back to this point next.

For the time being, we have in mind to justify more rigorously the conjectured equation. To do so, we propose the following argument based on a simple model for propagation of chaos.

2.2. On the Derivation of the limit equation in a toy model. To get an idea of the behavior of particles in our setting, we consider the simpler model in which the processes $(X^i)_i$ satisfy the following system of equations:

$$(5) \quad dX_t^i = \frac{1}{N} \sum_{j=1}^N \alpha^{i,j} X_t^j dt + dW_t^i, \quad i = 1, \dots, N,$$

where $(\alpha^{i,j})_{i,j=1,\dots,N}$ is a family of i.i.d. bounded random variables and $(W^i)_i$ is a family of independent standard Brownian motions. The families $(\alpha^{i,j})_{i,j}$ and $(W^i)_i$ are assumed to be independent. For simplicity, we do not address the influence of the threshold and do as if X_F was equal to $+\infty$. Also, we take the same deterministic initial conditions $X_0^i = x_0$ for all the particles.

We use the technique of coupling introduced in [24]. That is, we want to show convergence of (5) to

$$(6) \quad d\bar{X}_t = \mathbb{E}[\alpha] \mathbb{E}[\bar{X}_t] dt + dW_t, \quad \bar{X}_0 = x_0,$$

where α is another random variable with the same distribution as each $\alpha^{i,j}$ and W is a standard Brownian. Although it is not needed to state the equation, we assume for convenience that α and W are independent and that (α, W) is independent of the family $((\alpha_{i,j})_{i,j}, (W_i))$. To prove the convergence stated above, let us also introduce, for each $i \in \{1, \dots, N\}$, the SDE:

$$(7) \quad d\bar{X}_t^i = \mathbb{E}[\alpha] \mathbb{E}[\bar{X}_t^i] dt + dW_t^i, \quad \bar{X}_0^i = x_0,$$

and let us follow the strategy consisting in comparing (6) with (7) and (7) with (5).

As for (6) with (7), we notice that \bar{X} and \bar{X}^i have the same law. We therefore just need to consider X^i and \bar{X}^i . For the latter point, we observe that the processes $(\bar{X}, (\bar{X}^i)_i)$ are independent of the family $(\alpha^{i,j})_{i,j}$. Using the independence of α and \bar{X}^i , we also know that

$$\begin{aligned} d(X_t^i - \bar{X}_t^i) &= \left(\frac{1}{N} \sum_{j=1}^N \alpha^{i,j} X_t^j - \mathbb{E}[\alpha \bar{X}_t^i] \right) dt \\ &= \left[\left(\frac{1}{N} \sum_{j=1}^N \alpha^{i,j} X_t^j - \frac{1}{N} \sum_{j=1}^N \alpha^{i,j} \bar{X}_t^j \right) + \left(\frac{1}{N} \sum_{j=1}^N \alpha^{i,j} \bar{X}_t^j - \mathbb{E}[\alpha \bar{X}_t^i] \right) \right] dt. \end{aligned}$$

Since $\alpha^{i,j}$ is bounded by a constant C , we may write

$$|X_t^i - \bar{X}_t^i| \leq \int_0^t \frac{C}{N} \sum_{j=1}^N |X_s^j - \bar{X}_s^j| ds + \int_0^t \left| \frac{1}{N} \sum_{j=1}^N \alpha^{i,j} \bar{X}_s^j - \mathbb{E}[\alpha \bar{X}_s^i] \right| ds.$$

By summing over i and dividing by N , also defining δ_t to be the function $\frac{1}{N} \sum_{i=1}^N |X_t^i - \bar{X}_t^i|$, we get

$$\delta_t \leq C \int_0^t \delta_s ds + \frac{1}{N^2} \int_0^t \sum_{i=1}^N \left| \sum_{j=1}^N (\alpha^{i,j} \bar{X}_s^j - \mathbb{E}[\alpha \bar{X}_s^i]) \right| ds.$$

By using Gronwall's lemma, we then deduce that

$$\delta_t \leq \frac{\exp(Ct)}{N^2} \int_0^t \sum_{i=1}^N \left| \sum_{j=1}^N (\alpha^{i,j} \bar{X}_s^j - \mathbb{E}[\alpha \bar{X}_s^i]) \right| ds.$$

Taking the expectation, we have the following:

$$\mathbb{E}[\delta_t] \leq \frac{\exp(Ct)}{N} \int_0^t \sum_{i=1}^N \left(\frac{1}{N} \mathbb{E} \left| \sum_{j=1}^N (\alpha^{i,j} \bar{X}_s^j - \mathbb{E}[\alpha \bar{X}_s^i]) \right| \right) ds.$$

Moreover, it is now useful to investigate the second moment of $(\alpha^{i,j} \bar{X}_s^j - \mathbb{E}[\alpha \bar{X}_s^i])$, which is a finite value C_2^2 . Hence, by independence of $(\alpha^{i,j})_{i,j}$ and $(\bar{X}^j)_j$, we get:

$$\mathbb{E}[\delta_t] \leq \frac{\exp(Ct)}{N} \int_0^t \sum_{i=1}^N \frac{C_2}{N^{1/2}} ds \leq \exp(Ct) \frac{C_2 t}{N^{1/2}}.$$

Since it holds that

$$\delta_t \geq W_1 \left(\frac{1}{N} \sum_{i=1}^N \delta_{X_t^i}, \frac{1}{N} \sum_{i=1}^N \delta_{\bar{X}_t^i} \right),$$

where $W_1(\cdot, \cdot)$ is the 1-Wasserstein distance defined as $W_1(\mu, \nu) = \inf_{\pi} \int_{\mathbb{R}^2} |x - y| d\pi(x, y)$, the infimum being taken over all the probability measures π on \mathbb{R}^2 that have μ and ν as marginal distributions, we have also proved that, for any $T > 0$,

$$\lim_{N \rightarrow \infty} \sup_{0 \leq t \leq T} \mathbb{E} \left[W_1 \left(\frac{1}{N} \sum_{i=1}^N \delta_{X_t^i}, \frac{1}{N} \sum_{i=1}^N \delta_{\bar{X}_t^i} \right) \right] = 0.$$

By the law of large numbers, this implies that

$$\lim_{N \rightarrow \infty} \sup_{0 \leq t \leq T} \mathbb{E} \left[W_1 \left(\frac{1}{N} \sum_{i=1}^N \delta_{X_t^i}, \mathcal{L}(\bar{X}_t) \right) \right] = 0,$$

which is nothing but propagation of chaos.

2.3. Blow-up Argument. We now address the blow-up phenomenon for (4). Since (4) is similar, up to the scaling factor p , to the equation investigated in [1, 6], we expect the same picture: If the initial condition is too close to $X_F = 1$, then a blow-up occurs; if it is sufficiently far away, then it cannot occur. Here we show that a blow-up indeed exists if $X_0 = x_0$ for some deterministic x_0 that is close enough to 1. Our proof is based upon the probabilistic approach, but, in fact, it consists of a reformulation of the arguments used by Cáceres et al. in [1]. Below, we use repeatedly the notation $e(t) := \mathbb{E}[M_t]$.

Theorem 1. *Assume that drift coefficient satisfies, for some $\lambda > 0$,*

$$b(v) \geq -\lambda v, \quad \text{for all } -\infty < v \leq 1.$$

If the (deterministic) initial condition $X(0) = x_0 < 1$ is sufficiently close to the threshold 1, there is no global in time solution of the SDE

$$\begin{aligned} dX_t &= b(X_t) dt + \beta p e'(t) dt + dW_t - dM_t, \quad t \geq 0, \\ \tau_{k+1} &= \inf\{t \geq \tau_k, X_t \geq 1\} \\ M_t &= \sum_{k \geq 1} \mathbb{1}_{[0, t]}(\tau_k) \end{aligned} \tag{8}$$

with $e(t) = \mathbb{E}[M_t]$.

Proof. The idea is to retrace the proof in [1]. Assume indeed that we have a global in time solution $(X_t)_{t \geq 0}$ to (8) with a differentiable mean counter e . The object of our study is then $F_\mu(t) := \mathbb{E}[\varphi(X_t)]$ with $\varphi(x) = \exp(\mu x)$, where

$$X_t = X_0 + \int_0^t (b(X_s) + \beta p e'(s)) ds + W_t - M_t.$$

Now we apply Itô's formula to $\varphi(X_t)$:

$$\begin{aligned} \varphi(X_t) &= \varphi(X_0) + \int_0^t \underbrace{\varphi'(X_s)}_{\mu \varphi(X_s)} (b(X_s) + \alpha \cdot p e'(s)) ds + \int_0^t \underbrace{\varphi'(X_s)}_{\mu \varphi(X_s)} dW_s + \frac{1}{2} \int_0^t \underbrace{\varphi''(X_s)}_{\mu^2 \varphi(X_s)} ds \\ &\quad + \int_0^t \underbrace{[\varphi(X_{s-}) - 1] - \varphi(X_{s-})}_{(\varphi(0) - \varphi(1))} dM_s. \end{aligned}$$

Taking expectation:

$$\mathbb{E}[\varphi(X_t)] = \mathbb{E}[\varphi(X_0)] + \mu \int_0^t \mathbb{E}[\varphi(X_s)(b(X_s) + \beta \cdot p e'(s))] ds + \frac{\mu^2}{2} \int_0^t \mathbb{E}[\varphi(X_s)] ds + (\varphi(0) - \varphi(1))e(t).$$

Recalling that $F_\mu(t) = \mathbb{E}[\varphi(X_t)]$, we can rewrite the above expression as:

$$F_\mu(t) = F_\mu(0) + \mu \int_0^t \mathbb{E}[\varphi(X_s)(b(X_s) + \beta \cdot p e'(s))] ds + \frac{\mu^2}{2} \int_0^t F_\mu(s) ds + (\varphi(0) - \varphi(1))e(t).$$

Then, using the hypothesis on b , we get that

$$\begin{aligned} F_\mu(t) &\geq F_\mu(0) + \mu \int_0^t \mathbb{E}[\varphi(X_s)(\beta \cdot p e'(s) - \lambda X_s)] ds + \frac{\mu^2}{2} \int_0^t F_\mu(s) ds + (\varphi(0) - \varphi(1))e(t) \\ &\geq F_\mu(0) + \mu \int_0^t (\beta \cdot p e'(s) - \lambda) F_\mu(s) ds + \frac{\mu^2}{2} \int_0^t F_\mu(s) ds + (\varphi(0) - \varphi(1))e(t), \end{aligned}$$

that is

$$(9) \quad F_\mu(t) \geq F_\mu(0) + \int_0^t \mu \left(\beta \cdot p e'(s) - \lambda + \frac{\mu}{2} \right) F_\mu(s) ds + \left(\varphi(0) - \varphi(1) \right) e(t).$$

Define $\tilde{\lambda}$ as

$$\tilde{\lambda} := \frac{\varphi(1) - \varphi(0)}{\mu \beta p}$$

and let us choose μ such that $-\lambda + \frac{\mu}{2} > 0$. We can now proceed by stating that:

$$F_\mu(t) \geq F_\mu(0) + p \left(\int_0^t \mu \beta e'(s) F_\mu(s) ds - \tilde{\lambda} \beta \mu e(t) \right) \geq F_\mu(0) + p \int_0^t \mu \beta e'(s) [F_\mu(s) - \tilde{\lambda}] ds.$$

Claim: If $F_\mu(0) \geq \tilde{\lambda}$, then $F_\mu(t) \geq \tilde{\lambda}$. It suffices to define the first time $T := \inf\{t \geq 0 : F_\mu(t) < \tilde{\lambda}\}$ and to observe from the above inequality that $T = +\infty$.

Coming back to (9) we get that

$$\begin{aligned} F_\mu(t) &\geq F_\mu(0) + \int_0^t \mu \left(\beta p e'(s) - \lambda + \frac{\mu}{2} \right) F_\mu(s) ds - p \tilde{\lambda} \beta \mu e(t) \\ &\geq F_\mu(0) + p \int_0^t \mu \beta e'(s) F_\mu(s) ds + \int_0^t \mu \left(-\lambda + \frac{\mu}{2} \right) F_\mu(s) ds - p \tilde{\lambda} \beta \mu e(t) \\ &\geq F_\mu(0) + p \tilde{\lambda} \beta \mu e(t) + \int_0^t \mu \left(-\lambda + \frac{\mu}{2} \right) F_\mu(s) ds - p \tilde{\lambda} \beta \mu e(t) \\ &= F_\mu(0) + \int_0^t \mu \left(-\lambda + \frac{\mu}{2} \right) F_\mu(s) ds. \end{aligned}$$

This implies that:

$$(10) \quad F_\mu(t) \geq \exp \left(\mu \left(-\lambda + \frac{\mu}{2} \right) t \right) F_\mu(0).$$

But we know that

$$(11) \quad F_\mu(t) \leq \exp(\mu t).$$

From (10) and (11) we get a contradiction in the following sense: If we have a global in time solution $(X_t)_{t \geq 0}$ to (8) with a differentiable mean counter e and with an initial condition that satisfies $F_\mu(0) \geq \tilde{\lambda}$, then we get (10) and (11). As the latter two cannot be true at the same time, we deduce that, whenever $F_\mu(0) \geq \tilde{\lambda}$, there is no global in time solution to (8). \square

Remark 2. In summary, for a fixed (β, p) the phenomena of blow up holds for initial conditions that are concentrated around the threshold, namely for initial conditions such that

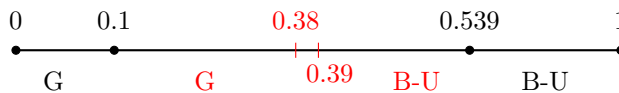
$$\exists \mu > 2\lambda \text{ such that } F_\mu(0) \geq \tilde{\lambda} \text{ with } \tilde{\lambda} = \frac{\varphi(1) - \varphi(0)}{\mu \beta p}.$$

Since we assumed for simplicity that the initial condition is deterministic, we know that $F_\mu(0) = \exp(\mu x_0)$. So, we have a blow-up for all p such that:

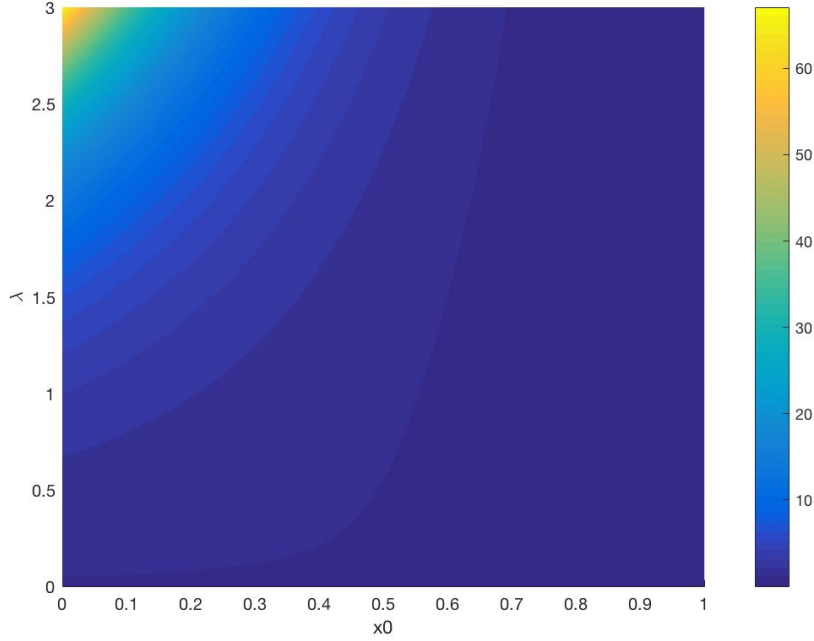
$$p \geq \inf_{\mu \geq 2\lambda} \frac{e^\mu - 1}{\mu \beta e^{\mu x_0}} =: p_c^+(x_0)$$

In Figure 1 we present the upper bound for the threshold p_c^+ , varying on the parameter of the drift λ and the initial condition x_0 . The parameter β is taken equal to 1. Of course, values of $p_c^+(x_0)$ that are above 1 are irrelevant, in the sense that there cannot be any blow-up in those cases.

Blow up discussion for the Brownian case. Let us focus on a particular case, in absence of drift, so $\lambda = 0$, with deterministic initial condition $x_0 = 0.8$ and $\beta = 1$. From the previous remark, we get that $p_c^+ \sim 0.539$. Moreover by Theorem 5.2 in [6] we get that $p_c^- \sim 0.1$, where p_c^- is a lower bound under which there is no blow-up. From simulations, we conjecture that there exists in fact a critical threshold $p_c(x_0) \in (p_c^-(x_0), p_c^+(x_0))$ under which there is no blow-up and above which there is a blow up; numerically, the threshold is around 0.385 (in red the result obtained by numerics).



In the more general case, where $\beta \neq 1$, we get that the threshold for p is simply rescaled by the factor β .


 FIGURE 1. Plot of the upper bound estimate p_c^+ for the threshold p_c

2.4. Dependent Random Weights. In this section, we will present a short analysis on the behavior of the network of neurons when the synaptic weights are dependent. In particular, we will focus on two different forms of random connections.

First, we focus on the case when $\alpha^{i,j} = \alpha^i \alpha^j$, where $(\alpha^i)_{i=1,\dots,N}$ are independent Bernoulli random variables with the same parameter p , in which case the dynamics of the system reads (this is case (2) in Subsection 1.3):

$$(12) \quad X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j M_t^j - M_t^i + W_t^i, \quad i = 1, \dots, N.$$

Our first purpose is to find out the limit equation. We conjecture that the limit equation for this system is the following:

$$(13) \quad X_t = X_0 + \int_0^t b(X_s) ds + \alpha \mathbb{E}[\alpha M_t] - M_t + W_t,$$

where α is another Bernoulli random variable of parameter p and W is an independent Brownian motion. The initial condition X_0 has the same distribution as any of the X_0^i 's and is independent of the pair (α, W) . We guess that (13) is the limit equation. Importantly, (13) does no longer read as a rescaled version of (2). The reason is the processes X^i and M^i should not become independent of α^i , even if the network size tends to infinity. This is the main rationale for investigating this example.

Equation (13), in fact, defines a network that consists of two different components: The first corresponds to those neurons for which $\alpha = 1$; it is a complete sub-network, composed by neurons which are all connected among themselves, so that neurons of this sub-network feel an interaction with all the other neurons of the same sub-network; The second component is made of isolated neurons for which $\alpha = 0$. As such, the limit equation catches out the duality of the population and the parameter p describes the relative size of the connected sub-network within the whole population. Noticeably, neurons in the asymptotic connected sub-network evolve according to a rescaled version of (2), see Remark 3 right below.

The second case we would like to deal with is the one with $\alpha^{i,j} = p \alpha^i$, where again $(\alpha^i)_i$ is an i.i.d. family of Bernoulli random variables of parameter p (this is case (3) in Subsection 1.3). The rationale for this choice is that $\alpha^{i,j}$ here reads as the limit of $\frac{1}{N} \alpha^i \sum_{j=1}^N \alpha^j$, which prompts us below to compare this example with the previous one. In this case as well, we have a network of neurons with two different populations: one formed by isolated neurons and the other one composed by neurons that are interact with one another with the same deterministic synaptic weight p . The dynamics of the particle system reads:

$$(14) \quad X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{p\alpha^i}{N} \sum_{j=1}^N M_t^j - M_t^i + W_t^i, \quad i = 1, \dots, N.$$

Using the fact that $p = \mathbb{E}[\alpha]$, our guess is that the limit mean field equation should be given by:

$$(15) \quad X_t = X_0 + \int_0^t b(X_s) ds + \alpha \mathbb{E}[\alpha] \mathbb{E}[M_t] - M_t + W_t.$$

Remark 3. In this remark, we try to make a first comparison between (13) and (15). As for (13), we observe that

$$\alpha \mathbb{E}[\alpha M_t] = \alpha \mathbb{E}[M_t \cdot \mathbb{1}_{\{\alpha=1\}}] = \alpha p \cdot \mathbb{E}[M_t \mid \alpha = 1],$$

which really says that, on the event $\alpha = 1$, (13) behaves like (4).

Regarding (15), we have:

$$(16) \quad \begin{aligned} \alpha p \cdot \mathbb{E}[M_t] &= \alpha p \left(\mathbb{E}[M_t \mid \mathbb{1}_{\{\alpha=1\}}] \cdot p + \mathbb{E}[M_t \mid \mathbb{1}_{\{\alpha=0\}}] \cdot (1-p) \right) \\ &= \alpha p^2 \cdot \mathbb{E}[M_t \mid \alpha = 1] + \alpha p(1-p) \cdot \mathbb{E}[M_t \mid \alpha = 0]. \end{aligned}$$

As we expect the activity of the isolated neurons to be less than the activity of the connected ones, our guess is that the leading term in (16) for causing a blow up is

$$\alpha p^2 \cdot \mathbb{E}[M_t \mid \alpha = 1].$$

So if we denote by p_c^1 the critical value of p (that should depend on the initial condition) for the occurrence of a blow-up in (13) and similarly by p_c^2 the critical value in (15), we can expect that $p_c^2 \approx \sqrt{p_c^1}$. Of course, we make the conjecture, based on the final discussion of the previous section, that there is indeed, for each model, a critical value of the connectivity p under which there is no blow-up and above which there is a blow-up.

2.5. Derivation of the limit equations in the toy model. Following Subsection 2.2, we here justify the mean field term manifesting in (13) by focusing on a toy example. To make it clear, instead of studying rigorously the limit behavior of

$$X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j M_t^j - M_t^i + W_t^i$$

(where α^i are i.i.d. Bernoulli variables with parameter p), we consider the simpler model introduced in Subsection 2.2:

$$dX_t^i = \frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j X_t^j dt + dW_t^i,$$

where $X_0^i = x_0$ is deterministic. We also introduce, following the notation already used, the processes \bar{X}^i described by

$$d\bar{X}_t^i = \alpha^i \mathbb{E}[\alpha^i \bar{X}_t^i] dt + dW_t^i.$$

Assuming that X^i and \bar{X}^i have the same initial conditions, we get

$$X_t^i - \bar{X}_t^i = \int_0^t \frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j (X_s^j - \bar{X}_s^j) ds + \int_0^t \left(\frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \alpha^i \mathbb{E}[\alpha^i \bar{X}_s^i] \right) ds.$$

Taking absolute values, multiplying by α^i , summing over i and dividing by N , we can assert that

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \alpha^i |X_t^i - \bar{X}_t^i| &\leq \int_0^t \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 \right) \left(\frac{1}{N} \sum_{i=1}^N \alpha^i |X_s^i - \bar{X}_s^i| \right) ds \\ &\quad + \int_0^t \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 \right) \left| \frac{1}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \mathbb{E}[\alpha^i \bar{X}_s^i] \right| ds. \end{aligned}$$

Defining $\tilde{\delta}_t = \frac{1}{N} \sum_{i=1}^N \alpha^i |X_t^i - \bar{X}_t^i|$, we can write that

$$\tilde{\delta}_t \leq \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 \right) \left[\int_0^t \tilde{\delta}_s ds + \int_0^t \left| \frac{1}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \mathbb{E}[\alpha^i \bar{X}_s^i] \right| ds \right].$$

By using Gronwall's lemma, we obtain

$$\tilde{\delta}_t \leq \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 \right) \exp \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 t \right) \int_0^t \left| \frac{1}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \mathbb{E} [\alpha^i \bar{X}_s^i] \right| ds.$$

Also defining $\delta_t = \mathbb{E}[\tilde{\delta}_t]$, $A = \left(\frac{1}{N} \sum_{i=1}^N (\alpha^i)^2 \right)$, taking expectation and using Cauchy-Schwarz inequality, we have

$$\delta_t \leq \left(\mathbb{E} [A^2 \exp (2At)] \right)^{1/2} \int_0^t \left(\mathbb{E} \left[\left(\frac{1}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \mathbb{E} [\alpha^i \bar{X}_s^i] \right)^2 \right] \right)^{1/2} ds.$$

We can manage the last term in the expression above as we did before. In fact, using a standard independence argument, we get

$$\mathbb{E} \left[\left(\frac{1}{N} \sum_{j=1}^N \alpha^j \bar{X}_s^j - \mathbb{E} [\alpha^i \bar{X}_s^i] \right)^2 \right] = \frac{1}{N} \left(\mathbb{E} [(\alpha^1 \bar{X}_s^1)^2] - (\mathbb{E} [\alpha^1 \bar{X}_s^1])^2 \right).$$

Therefore,

$$\delta_t \leq \left(\mathbb{E} [A^2 \exp (2At)] \right)^{1/2} \cdot \frac{1}{N^{1/2}} \int_0^t \left(\mathbb{E} [(\alpha^1 \bar{X}_s^1)^2] - (\mathbb{E} [\alpha^1 \bar{X}_s^1])^2 \right)^{1/2} ds,$$

and we conclude as in Subsection 2.2. The derivation of the toy version of equation (15) from the corresponding toy version of the network of interacting neurons (14) follows from a similar argument.

2.6. Blow-up Argument. We now provide analogs to Theorem 1 for (13) and (15). Let us first consider the second case, i.e., $\alpha^{i,j} = p\alpha^i$, with α^i Bernoulli independent random variables of parameter p .

Theorem 4. Assume that drift coefficient satisfies, for some $\lambda > 0$,

$$b(v) \geq -\lambda v, \quad \text{for all } -\infty < v \leq 1.$$

If the (deterministic) initial condition $X(0) = x_0 < 1$ is sufficiently near the threshold 1, there is no global in time solution of the SDE

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha p \cdot \mathbb{E} [M_t] + W_t - M_t,$$

where $\alpha \sim \mathcal{B}(p)$ is independent of W .

Proof. The idea of the proof is very close to that of Theorem 1. We want to find an initial condition that leads to a contradiction if we assume that $t \mapsto e(t) = \mathbb{E}(M_t)$ is differentiable. However, the proof slightly differs since we here focus on the quantity $F_\mu^\alpha(t) := \mathbb{E}[\alpha \varphi(X_t)]$. Assuming that e is indeed differentiable, applying Itô's formula to $(\varphi(X_t))_{t \geq 0}$, multiplying by α and taking the expectation, we get

$$F_\mu^\alpha(t) = F_\mu^\alpha(0) + \int_0^t \mathbb{E} \left[\mu \alpha \varphi(X_s) \left(-\lambda + \alpha p e'(s) + \frac{\mu}{2} \right) \right] ds + \left(\varphi(0) - \varphi(1) \right) \mathbb{E}[\alpha M_t].$$

The random variable α takes values in $\{0, 1\}$ and M_t is non-negative, which makes it possible to state that $\mathbb{E}[\alpha M_t] \leq \mathbb{E}[M_t]$. Therefore, since $(\varphi(0) - \varphi(1))$ is negative,

$$(17) \quad F_\mu^\alpha(t) \geq F_\mu^\alpha(0) + \int_0^t \mu \left(-\lambda + \frac{\mu}{2} \right) F_\mu^\alpha(s) ds + \int_0^t \mu p e'(s) F_\mu^\alpha(s) ds + \left(\varphi(0) - \varphi(1) \right) e(t).$$

Now, the proof exactly matches that of Theorem 1. Choose indeed $-\lambda + \frac{\mu}{2} > 0$ and take the initial condition close enough to the threshold 1 so that

$$F_\mu^\alpha(0) \leq \tilde{\lambda} \quad \text{with} \quad \tilde{\lambda} = \frac{1}{\mu p} (-\varphi(0) + \varphi(1)).$$

We then get

$$F_\mu^\alpha(t) \geq F_\mu^\alpha(0) \exp \left(\mu \left(-\lambda + \frac{\mu}{2} \right) t \right), \quad t > 0.$$

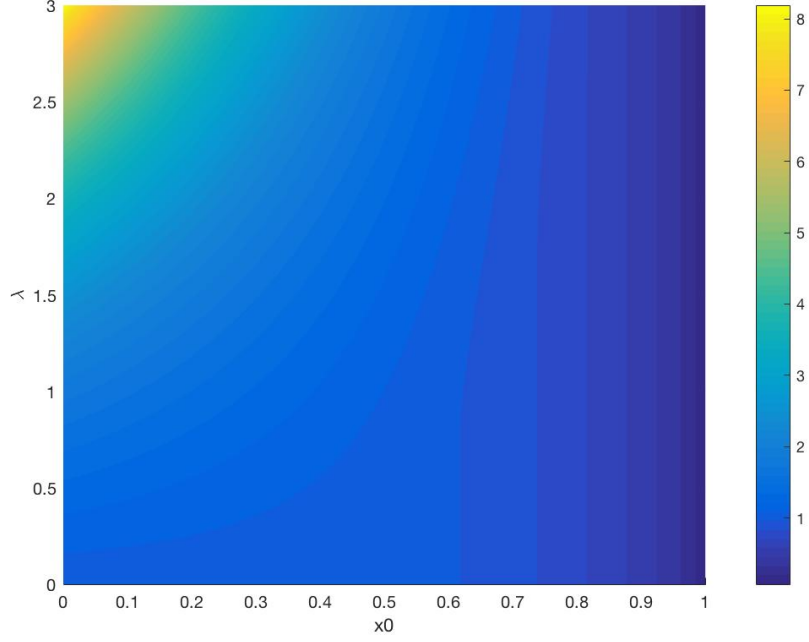
This is a contradiction because $X_t \leq 1$ and $\alpha \in \{0, 1\}$, which means that $F_\mu^\alpha(t) \leq e^\mu$. □

Remark 5. As explained in Remark 2, there is blow-up when

$$\exists \mu > 2\lambda \text{ such that } F_\mu(0) \geq \tilde{\lambda} \text{ with } \tilde{\lambda} = \frac{\varphi(1) - \varphi(0)}{\mu p}.$$

Assuming for simplicity that the initial condition is deterministic, we get:

$$F_\mu^\alpha(0) = p \cdot \exp(\mu x_0)$$


 FIGURE 2. Plot of the upper bound estimate $p_c^{+,2}$ for the threshold p_c

So, given a deterministic initial condition, we have a blow-up when p is such that:

$$p \geq \sqrt{\inf_{\mu \geq 2\lambda} \frac{e^\mu - 1}{\mu e^{\mu x_0}}} =: p_c^{+,2}(x_0)$$

In Figure 2, we present the upper bound for the threshold $p_c^{+,2}$, varying on the parameter of the drift λ and the initial condition x_0 .

Consider now the case $\alpha^{i,j} = \alpha^i \alpha^j$, with α^i Bernoulli independent random variables of parameter p .

Theorem 6. Assume that drift coefficient satisfies, for some $\lambda > 0$,

$$b(v) \geq -\lambda v, \quad \text{for all } -\infty < v \leq 1.$$

If the (deterministic) initial condition $X(0) = x_0 < 1$ is sufficiently near the threshold 1, there is no global in time solution of the SDE

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha \cdot \mathbb{E}[\alpha M_t] + W_t - M_t,$$

where $\alpha \sim \mathcal{B}(p)$ is independent of W .

Proof. The strategy is almost the same as in the previous case, namely the main tool is the function $F_\mu^\alpha(t) := \mathbb{E}[\alpha \varphi(X_t)]$. By Itô's Formula:

$$F_\mu^\alpha(t) = F_\mu^\alpha(0) + \int_0^t \mathbb{E} \left[\mu \alpha \varphi(X_s) \left(-\lambda + \alpha e'_\alpha(s) + \frac{\mu}{2} \right) \right] ds + \left(\varphi(0) - \varphi(1) \right) \mathbb{E}[\alpha M_t].$$

with $e_\alpha(t) = \mathbb{E}[\alpha M_t]$. With the same calculation as in the previous case, we get

$$F_\mu^\alpha(t) \geq F_\mu^\alpha(0) + \int_0^t \mu \left(-\lambda + \frac{\mu}{2} \right) F_\mu^\alpha(s) ds + \int_0^t \mu e'_\alpha(s) F_\mu^\alpha(s) ds + \left(\varphi(0) - \varphi(1) \right) e_\alpha(t).$$

The only differences with respect to the previous case are that we have $e_\alpha(t)$ instead of $e(t)$ and that there is no p in the third term. \square

Remark 7. As before, there is blow-up when

$$\exists \mu > 2\lambda \text{ such that } F_\mu^\alpha(0) \geq \tilde{\lambda} \text{ with } \tilde{\lambda} = \frac{\varphi(1) - \varphi(0)}{\mu}$$

Assuming for simplicity that the initial condition is deterministic, we get:

$$F_\mu(0) = p \cdot \exp(\mu x_0).$$

So, given a deterministic initial condition, we get a blow-up when p is such that:

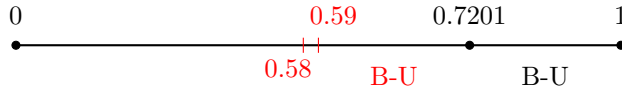
$$p \geq \inf_{\mu \geq 2\lambda} \frac{e^\mu - 1}{\mu e^{\mu x_0}} =: p_c^{+,1}(x_0),$$

which fits the formula obtained in Remark 2. So the upper bound varying on the parameters (x_0, λ) is the same of Figure 1. Observe also that

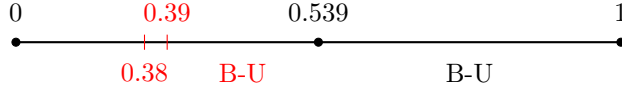
$$p_c^{+,2}(x_0) = \sqrt{p_c^{+,1}(x_0)},$$

which is consistent with our informal discussion in Remark 3.

Blow up discussion for the Brownian case. Let us focus on the case where the network includes a non complete subnetwork following the dynamics prescribed in (15) (or equivalently case (3) in Subsection 1.3) in absence of drift, i.e. $b \equiv 0$, with deterministic initial condition $x_0 = 0.8$. From the previous remark, we get that $p_c^{+,2} \sim 0.7201$. Through simulations, we look for the threshold value p_c (with the same conjecture as before that, above this threshold, there is a blow-up whilst there is no blow-up below the threshold). Here is the plot that we get (numerical values are in red):



Recall from the discussion right after Remark 2 the plot we get for the network of type (13) (or equivalently case (2) in Subsection 1.3):



Notice that the numerical results confirm what we predict in Remarks 3, 5, 7; observe in particular that $\sqrt{0.39} \sim 0.62$, which not so far from 0.59.

2.7. Comparison between the two models. In this section we continue to investigate the differences between the two cases (2) and (3) in Subsection 1.3. We here consider the following SDEs:

$$X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\alpha^i}{N} p \sum_{j=1}^N M_t^j + W_t^i - M_t^i,$$

$$X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\alpha^i}{N} \sum_{j=1}^N \alpha^j M_t^j + W_t^i - M_t^i,$$

with the usual definitions. Recall that our guess is that the two systems converge to the limit SDEs (compare with (13) and (15)):

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha p \mathbb{E}[M_t] + W_t - M_t,$$

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha \mathbb{E}[\alpha M_t] + W_t - M_t,$$

again with the usual definitions. Using the same conditioning argument as in Remark 3, these two last SDEs can easily be rewritten as:

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha p \left(p \cdot \mathbb{E}[M_t | \alpha = 1] + (1-p) \cdot \mathbb{E}[M_t | \alpha = 0] \right) + W_t - M_t,$$

$$X_t = X_0 + \int_0^t b(X_s) ds + \alpha \left(p \cdot \mathbb{E}[M_t | \alpha = 1] \right) + W_t - M_t.$$

Assuming, as we did to compute thresholds numerically, that the initial condition is deterministic, i.e., $X_0 = x_0$, and that b is zero, we finally get:

$$(18) \quad X_t = x_0 + \left(\alpha p^2 \cdot \mathbb{E}[M_t | \alpha = 1] + \alpha p(1-p) \cdot \mathbb{E}[M_t | \alpha = 0] \right) + W_t - M_t,$$

$$(19) \quad X_t = x_0 + \left(\alpha p \cdot \mathbb{E}[M_t | \alpha = 1] \right) + W_t - M_t.$$

We now study equation (18) and compare it to (19). First, let us consider the term $\mathbb{E}[M_t | \alpha = 0]$: this is the expectation of the process M_t given the fact that the neuron is not concerned by interaction. The good point is

that the limiting model without interaction can be solved explicitly. Namely, we can identify $\mathbb{E}[M_t | \alpha = 0]$ with $\mathbb{E}[\tilde{M}_t]$, where \tilde{M}_t solves the SDE:

$$\tilde{X}_t + \tilde{M}_t = x_0 + \tilde{W}_t, \quad \tilde{M}_t = \left[\left(\sup_{s \in [0, t]} (\tilde{X}_s + \tilde{M}_s) \right) \right]_+ = \left[\left(x_0 + \sup_{s \in [0, t]} \tilde{W}_s \right) \right]_+,$$

where $\lfloor y \rfloor$ is the floor part of y . Calling f_t the density of the running maximum of Brownian motion until time t , we get:

$$\mathbb{E}[\tilde{M}_t] = \int_{\mathbb{R}} \lfloor (x_0 + x)_+ \rfloor f_t(x) dx.$$

Recalling that $f_t(x) = [\sqrt{2/\pi t} \exp(-x^2/2t)] \mathbb{1}_{\{x \geq 0\}}$, the above expression becomes

$$\begin{aligned} \mathbb{E}[\tilde{M}_t] &= \sqrt{\frac{2}{\pi t}} \int_{\mathbb{R}^+} \lfloor (x_0 + x)_+ \rfloor \exp\left(-\frac{x^2}{2t}\right) dx \\ &= \sqrt{\frac{2}{\pi t}} \int_{-x_0 \vee 0}^{+\infty} \lfloor (x_0 + x)_+ \rfloor \exp\left(-\frac{x^2}{2t}\right) dx \\ &= \frac{4}{\sqrt{\pi}} \int_{(-x_0/\sqrt{2t}) \vee 0}^{+\infty} \left\lfloor \left(x_0 + \sqrt{2t}x \right)_+ \right\rfloor \exp(-x^2) dx \\ &= \frac{4}{\sqrt{\pi}} \sum_{k \in \mathbb{N}} k \int_{((-x_0+k)/\sqrt{2t}) \vee 0}^{((-x_0+k+1)/\sqrt{2t}) \vee 0} \exp(-x^2) dx. \end{aligned}$$

Particularly, since $x_0 < 1$, it holds that $-x_0 + k$ is always positive for $k \geq 1$: therefore, $(-x_0 + k)/\sqrt{2t} \vee 0 = (-x_0 + k)/\sqrt{2t}$. Calling $e_0(t) := \mathbb{E}[\tilde{M}_t]$, we then see that the function e_0 is differentiable and its derivative is

$$e'_0(t) = \frac{1}{t} \sqrt{\frac{2}{\pi t}} e^{-\frac{x_0^2}{2t}} \sum_{k \in \mathbb{N}} k \left[(-x_0 + k) e^{-\frac{-k^2 + 2x_0 k}{2t}} - (-x_0 + k + 1) e^{-\frac{-(k+1)^2 + 2x_0(k+1)}{2t}} \right].$$

Since $x_0 < 1$, we have $-k^2 + 2x_0 k \geq 0$ whenever $k \geq 2$. Hence, we obtain the following:

$$\begin{aligned} e'_0(t) &\leq \frac{1}{t} \sqrt{\frac{2}{\pi t}} e^{-\frac{x_0^2}{2t}} \left[(1 - x_0) \left[e^{-\frac{-1+2x_0}{2t}} - e^{-\frac{-2(1-x_0)}{t}} \right] + \right. \\ &\quad \left. + \sum_{k \geq 2} k \left[(-x_0 + k) e^{-\frac{-k^2 + 2x_0 k}{2t}} - (-x_0 + k + 1) e^{-\frac{-(k+1)^2 + 2x_0(k+1)}{2t}} \right] \right], \end{aligned}$$

and then:

$$e'_0(t) \leq \frac{1}{t} \sqrt{\frac{2}{\pi t}} (1 - x_0) e^{-\frac{(1-x_0)^2}{2t}} + \frac{1}{t} \sqrt{\frac{2}{\pi t}} e^{-\frac{x_0^2}{2t}} \sum_{k \geq 2} k \left[(-x_0 + k) e^{-\frac{-k^2 + 2x_0 k}{2t}} - (-x_0 + k + 1) e^{-\frac{-(k+1)^2 + 2x_0(k+1)}{2t}} \right],$$

that is, for a finite constant $C_{x_0, T}$,

$$\sup_{0 \leq t \leq T} e'_0(t) \leq C_{x_0, T}.$$

We now come back to (18) and (19). We can rewrite (18) in the form:

$$(20) \quad X_t = x_0 + \alpha p(1-p) \int_0^t e'_0(s) ds + \left(\alpha p^2 \cdot \mathbb{E}[M_t | \alpha = 1] \right) + W_t - M_t,$$

which shows that, in comparison with (19), there is not only an additional factor p in the conditional mean field term given the event $\alpha = 1$, but there is also an additional drift e'_0 . In other words, if we omit the term $\alpha p(1-p) \int_0^t e'_0(s) ds$ in (20), we recover (19), but with p replaced by p^2 . Intuitively, this says that, if e'_0 was equal to 0, we would expect $p_c^2 = \sqrt{p_c^1}$. In fact, e'_0 is positive: It helps pushing the particles towards the threshold. As a result, it makes sense to expect $p_c^2 < \sqrt{p_c^1}$. Fortunately, this is exactly what shows up in the numerical analysis performed at the end of Subsection 2.6.

2.8. Independent Bernoulli random variables with parameter p_N . We now address case (4) in Subsection 1.3, namely we deal with a neural network described by the equation

$$X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\beta}{p_N \cdot N} \sum_{j=1}^N \alpha_N^{i,j} M_t^j + W_t^i - M_t^i,$$

where $\alpha_N^{i,j}$ are i.i.d. Bernoulli random variables with parameter p_N **depending** on N , the total number of neurons in the network. We are interested in three cases:

- $p_N \cdot N = \log^{1/2}(N)$;
- $p_N \cdot N = \log(N)$;
- $p_N \cdot N = \log^2(N)$.

The reason why we choose these three parameters follows from the theory of Erdős-Renyi for random graphs, see [9, 10] for undirected graphs and [14, 23] for directed graphs. We indeed know that $p_N = \log(N)/N$ is a sharp threshold for connectedness, meaning that:

- If $p_N < (1 - \varepsilon) \log(N)/N$, then the probability that the graph has a unique connected component tends to 0 as N goes to infinity;
- $p_N > (1 + \varepsilon) \log(N)/N$, then the probability that the graph has a unique connected component tends to 1 as N goes to infinity.

In contrast with more standard particle systems like those addressed in [24] or (say) of the same form as in (5), it is here possible to have several neurons hitting the firing potential at the same instant of time, and, most of all, it is even possible to have a neuron spiking more than once at the same instant of time, which fact may be excluded in other models of neural networks, see for instance [6]. As exemplified in [4, 7], this requires a sequential definition of the spikes that may occur at the same time, given by an induction with the following initialization:

$$\Gamma_0 := \{i \in \{1, \dots, N\} : X_{t-}^i = 1\}.$$

We say that a spike occurs at time t if $\Gamma_0 \neq \emptyset$. Because of the interactions between the neurons, neurons in the set Γ_0 may force the others to jump at the same time t . This happens for neuron $i \notin \Gamma_0$ if

$$X_{1,t-}^i := X_{t-}^i + \frac{\beta}{p_N \cdot N} \sum_{j=1}^N \alpha_N^{i,j} \mathbb{1}_{\{j \in \Gamma_0\}} \geq 1,$$

which prompts us to let $\Gamma_1 := \{i \in \{1, \dots, N\} \setminus \Gamma_0 : X_{1,t-}^i \geq 1\}$. Iteratively, we define for any $i \in \{1, \dots, N\}$:

$$X_{k+1,t-}^i := \begin{cases} X_{k,t-}^i + \frac{\beta}{p_N \cdot N} \sum_{j=1}^N \alpha_N^{i,j} \mathbb{1}_{\{j \in \Gamma_k\}} & \text{if } i \notin \Gamma_k, \\ X_{k,t-}^i - \lfloor X_{k,t-}^i \rfloor & \text{if } i \in \Gamma_k, \end{cases}$$

$$\Gamma_{k+1} := \{i \in \{1, \dots, N\} \setminus \Gamma_k : X_{k+1,t-}^i \geq 1\}.$$

And then we let

$$(21) \quad \begin{aligned} M_t^i - M_{t-}^i &= \sum_{k \in \mathbb{N}} \mathbb{1}_{\Gamma_k}(i), \\ X_t^i &= \lim_{k \rightarrow \infty} X_{k,t-}^i \quad \text{if the limit exists.} \end{aligned}$$

Intuitively, Γ_k is the set of neurons that spike at the k th iteration, $X_{k,t-}^i$ is the potential of neuron i before the k th iteration and $X_{k+1,t-}^i$ is the potential after the k th iteration. With this rule we observe that:

- It is possible that one neuron receives (at a given iteration) a cumulative kick (from all the other particles) that is greater than 1, meaning that the potential of a neuron can jump from a potential less than 1 to a potential greater than 2. With our definition of $X_{k+1,t-}^i$ if $i \in \Gamma_k$, we just regard the whole as a single spike.

Still a neuron can spike several times at the same time:

- It is indeed possible that a neuron i spikes, that its kick makes others spike and that those, in return, make i spike again. This behavior may repeat again: when it repeats just for a finite number of times, we call it a “finite cascade”. In that case, the limit in (21) is well defined.
- It may happen that the cascading behavior just described above goes on infinitely many times: we call it an “infinite cascade”. In that case the limit in (21) may not exist.

Below, we choose $\beta < 1$. We then try to make a connection between neurons that spike more than twice and neurons that have a large degree, where, by definition, the degree d_i of neuron i is $d_i := \sum_{j=1}^N \alpha_N^{i,j}$. Our guess is based on the fact that, for a neuron that is connected to neurons that do not spike more than once at time t , the only way for it to record more than two spikes at time t is that its degree is higher than $\lceil p_N N / \beta \rceil$. In particular, the first neurons that record a second spike in the inductive construction of the sets $(\Gamma_k)_k$ must have a large degree. In this regard, we show below that the probability that a neuron has a high degree gets smaller and smaller as N tends to ∞ . We thus conjecture that, for the prescribed values of parameters, most of the neurons can only jump once at a given time and, henceforth, that, among those that record more than two spikes at the same time, most of them have a large degree.

We are therefore interested in computing

$$\mathbb{P} \left[\frac{1}{N} \sum_{j=1}^N \alpha_N^{1,j} \geq \frac{p_N}{\beta} \right],$$

where the apex 1 may be substituted by any other index i .

Remark 8 (Estimate on the upper bound for the probability to have a multiple spikes). *Observing that p_N/β is bigger than p_N and using the Cramer's Theorem (in the context of Large Deviation Theory), we find out that*

$$(22) \quad \mathbb{P} \left[\frac{1}{N} \sum_{j=1}^N \alpha_N^{1,j} \geq \frac{p_N}{\beta} \right] \leq e^{-N \cdot \Lambda^* \left(\frac{p_N}{\beta} \right)},$$

where

$$\Lambda^*(x) := x \log \left(\frac{x}{p_N} \right) + (1-x) \log \left(\frac{1-x}{1-p_N} \right).$$

We have that

$$(23) \quad -\log \left(e^{-N \cdot \Lambda^* \left(\frac{p_N}{\beta} \right)} \right) = \frac{1}{\beta} \log \left(\frac{1}{\beta} \right) \log^k(N) + \left(N - \frac{\log^k(N)}{\beta} \right) \log \left(\frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \right),$$

where k is such that $p_N = \log^k(N)/N$ (recall that $k = 1/2, 1, 2$ in the three typical cases we have in mind). The last part above is

$$\left(N - \frac{\log^k(N)}{\beta} \right) \log \left(\frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \right) = \left(N - \frac{\log^k(N)}{\beta} \right) \log \left(1 - \frac{1-\beta}{\beta} \frac{\log^k(N)}{N - \log^k(N)} \right).$$

Therefore, the last expression becomes

$$\left(N - \frac{\log^k(N)}{\beta} \right) \left(-\frac{1-\beta}{\beta} \frac{\log^k(N)}{N - \log^k(N)} + \mathcal{O} \left(\left(\frac{\log^k(N)}{N - \log^k(N)} \right)^2 \right) \right),$$

which can be rewritten as

$$\begin{aligned} & \frac{\beta N - \log^k(N)}{\beta} \left(-\frac{(1-\beta) \log^k(N)}{\beta(N - \log^k(N))} \right) + \frac{\beta N - \log^k(N)}{\beta} \cdot \mathcal{O} \left(\left(\frac{\log^k(N)}{N - \log^k(N)} \right)^2 \right) \\ &= \frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \left(-\frac{(1-\beta) \log^k(N)}{\beta} \right) \\ &+ \frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \cdot \frac{N - \log^k(N)}{\log^k(N)} \log^k(N) \cdot \mathcal{O} \left(\left(\frac{\log^k(N)}{N - \log^k(N)} \right)^2 \right). \end{aligned}$$

Finally, we can again rewrite the expression above as

$$\left(\frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \right) \left(-\frac{1-\beta}{\beta} + \mathcal{O} \left(\frac{\log^k(N)}{N - \log^k(N)} \right) \right) \log^k(N).$$

Now, the first term in the expression above converges to 1 when $N \rightarrow +\infty$; similarly, the “big O ” multiplied by $\log^k(N)$ goes to 0 when $N \rightarrow +\infty$; this means that, for every $c < 1$,

$$\left(\frac{\beta N - \log^k(N)}{\beta N - \beta \log^k(N)} \right) \left(-\frac{1-\beta}{\beta} + \mathcal{O} \left(\frac{\log^k(N)}{N - \log^k(N)} \right) \right) \log^k(N) \geq -c \left(\frac{1-\beta}{\beta} \right) \log^k(N),$$

for N large enough. Coming back to (23):

$$-\log \left(e^{-N \cdot \Lambda^* \left(\frac{p_N}{\beta} \right)} \right) \geq \left[\frac{1}{\beta} \left(\log \left(\frac{1}{\beta} \right) - c \right) + c \right] \log^k(N).$$

Calling $C(c, \beta) := \frac{1}{\beta} \left(\log \left(\frac{1}{\beta} \right) - c \right) + c > \frac{1}{\beta} - 1 - \frac{c}{\beta} + c = \left(\frac{1}{\beta} - 1 \right) (1 - c) > 0$, we deduce that

$$e^{-N \cdot \Lambda^* \left(\frac{p_N}{\beta} \right)} \leq e^{-C(c, \beta) \log^k(N)},$$

which is especially meaningful when $k > 0$. To conclude, we have found that, for N large enough,

$$(24) \quad \mathbb{P} \left[\frac{1}{N} \sum_{j=1}^N \alpha_N^{1,j} \geq \frac{p_N}{\beta} \right] \leq e^{-C(c, \beta) \log^k(N)}.$$

Our conjecture is that, for $k > 0$, this should be a good approximation of the probability to observe more than two spikes for a single neuron at the same time.

By letting c tend to 1 in (24), we get in fact the following large deviation upper bound.

Theorem 9. Assume that $\beta < 1$ and choose p_N in the form $p_N = \log^k(N)/N$. Then,

$$(25) \quad \limsup_{N \rightarrow \infty} a_N^{-1} \log \left(\mathbb{P} \left[\frac{1}{N} \sum_{j=1}^N \alpha_N^{1,j} \geq \frac{p_N}{\beta} \right] \right) \leq -C(\beta),$$

with $a_N = N \cdot p_N = \log^k(N)$ and $C(\beta) := \frac{1}{\beta} \left(\log \left(\frac{1}{\beta} \right) - 1 \right) + 1$.

Remark 10. As for the lower bound, we did not manage to implement the usual tilting method used in the proof of Cramer's theorem. The fact that $(a_N)_{N \geq 0}$ increases at a rate that is much smaller than the rate of convergence of the law of large numbers, as given by the central limit theorem, is a hindrance.

2.9. Comparison with the Mean Field Case. A crucial question is to decide whether the mean field approximation is still valid in each of the three cases addressed in the preliminary discussion of Subsection 2.8. Observe indeed that, since p_N is assumed to get smaller and smaller as N tends to 0, it is by no means clear that the arguments used in Subsections 2.2 and 2.5 still apply. So, we consider the network given by a family of i.i.d. Bernoulli synaptic weights $(\alpha_N^{i,j})_{i,j} \sim \mathcal{B}(p_N)$, with p_N as before, and we want to compare numerically the behavior of the particle system driven by the interaction term

$$(26) \quad \frac{\beta}{N \cdot p_N} \sum_{j=1}^N \alpha_N^{i,j} M_t^j$$

with the behavior of the particle system driven by the interaction term

$$(27) \quad \frac{1}{N} \sum_{j=1}^N \gamma^{i,j} M_t^j,$$

where $(\gamma^{i,j})_{i,j} \sim \mathcal{B}(\beta)$ are i.i.d. Bernoulli variables of parameter β . The comparison between both is motivated by the fact that $\mathbb{E}[\beta \alpha_N^{i,j}/p_N] = \beta$. And for sure, our preliminary investigations have shown that (27) had the same limit behavior as the model driven by the simpler interaction term

$$(28) \quad \frac{\beta}{N} \sum_{j=1}^N M_t^j.$$

We provide in the left pane in Figure 3) simulations of the interaction terms in both cases, (26) and (27), when $p_N = \log^2(N)/N$ and $\beta = 0.375$. It shows that, for N large, both are indeed close. This seems to be a numerical evidence that the mean field approximation should be valid when p_N is above the Erdős-Rényi threshold for connectedness of the interaction graph. When $p_N = \log(N)/N$ and $p_N = \log^{1/2}(N)/N$, and for the same value of β , the center and right panes in Figure 3 suggest that the mean field approximation is no longer valid below the Erdős-Rényi threshold. These observations seem to be in accordance with the results obtained in [8] for the mean field approximation of another interacting diffusion model (so-called Kuramoto model) on a random Erdős-Rényi graph: Therein, the mean field approximation is shown to hold true if $p_N N / \log(N) \rightarrow \infty$.

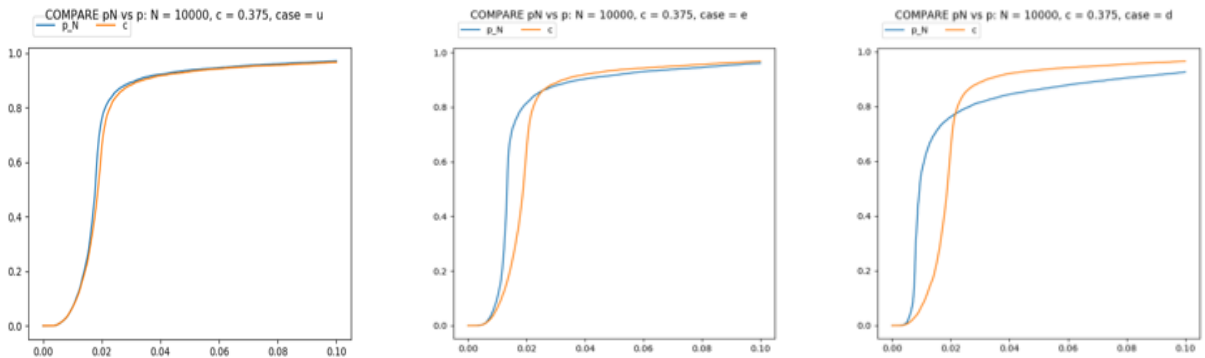


FIGURE 3. Comparison between interactions terms for $p_N = \log^2(N)/N$, $p_N = \log(N)/N$ and $p_N = \log^{1/2}(N)/N$.

3. ALGORITHM

3.1. Introduction. In this section, we present the algorithms used to simulate the particles system (3). We also consider a slightly different model in Section 3.3, where the neurons no longer spike when their membrane potential crosses a fixed deterministic threshold but they spike with *rate* $f^i(X_t^i)$ at time t . (see Section 3.3 for a precise definition).

Clock-driven vs. event-driven algorithm. The simulation of a very large network of neurons requires to carefully organize the program. We have used two different methods.

Clock-driven method. We fix a time step Δ and approximate the values of the membrane potentials at each time step $T_0 + \ell\Delta$ for $\ell \in \mathbb{N}$. This method is quite simple to implement, but not very fast. The choice of the time step Δ is crucial: it has to be very small compared to the typical length of an InterSpike Interval (ISI).

Event-driven method. This method consists in, first, simulating the times at which *events* occur, and second, updating the state of the network at these times. The advantage is that the clock is automatically fitted to the state of the system. In our algorithm, we have chosen the spiking times to describe the important *events* of this method.

We have used the *clock-driven method* to simulate the hard threshold model and the *event-driven method* for the soft threshold model. In this case, our algorithm is an extension of Ogata's algorithm, see [21].

3.2. Hard Threshold. We consider the finite system defined by (1), which we rewrite below for the sake of convenience:

$$(29) \quad \begin{cases} X_t^i = X_0^i + \int_0^t b(X_s^i) ds + \frac{\beta}{N} \sum_{j=1}^N \alpha^{i,j} M_t^j + W_t^i - (X_F - X_R) M_t^i, \\ M_t^i = \sum_{k \in \mathbb{N}} \mathbb{1}_{[0,t]}(\tau_k^i), \end{cases} \quad \forall i \in \{1, \dots, N\},$$

where, as usual, τ_k^i represents the time at which neuron i reaches the threshold $X_F > 0$ for the k -th time; above, X_R is the (common) reset values of the neurons.

In this section, we simulate the network at each time step $\ell\Delta$, where Δ is our fixed discretization parameter. The network is fully described by $((\bar{X}_{\ell\Delta}^i, \bar{M}_{\ell\Delta}^i), i \in \{1, \dots, N\}, \ell\Delta \leq T)$.

- Initialization $\ell = 0$.
 - We set $\bar{M}_0^i = 0$ and simulate the membrane potentials \bar{X}_0^i as independent and identically distributed according to the initial law.
- Step $\ell\Delta \rightarrow (\ell + 1)\Delta$.
 - Evolution of the membrane potentials without interaction, nor spontaneous spikes:

$$\hat{\bar{X}}_{(\ell+1)\Delta}^i = \bar{X}_{\ell\Delta}^i + b(\bar{X}_{\ell\Delta}^i)\Delta + W_{(\ell+1)\Delta}^i - W_{\ell\Delta}^i.$$

- Construction of the set $\Gamma_{(\ell+1)\Delta}^0$ of those neurons that are spiking first at time $(\ell + 1)\Delta$. First, set $\Gamma_{(\ell+1)\Delta}^0 := \{i \in \{1, \dots, N\}, \hat{\bar{X}}_{(\ell+1)\Delta}^i \geq X_F\}$. For each neuron $i \notin \Gamma_{(\ell+1)\Delta}^0$, we have $\bar{X}_{\ell\Delta}^i \leq X_F$ and $\hat{\bar{X}}_{(\ell+1)\Delta}^i \leq X_F$ but it could happen that $\sup_{\ell\Delta \leq t \leq (\ell+1)\Delta} \hat{\bar{X}}_t^i \geq X_F$, i.e. the process crosses the threshold during one time step but is back below the threshold at time $(\ell + 1)\Delta$. We approximate the probability of such an event by the well known corresponding probability for a Brownian bridge (see [13]), that is

$$p^i = \exp \left(-2 \frac{(X_F - \hat{\bar{X}}_{(\ell+1)\Delta}^i)(X_F - \bar{X}_{\ell\Delta}^i)}{\Delta} \right).$$

More precisely, for each $i \notin \Gamma_{(\ell+1)\Delta}^0$, we simulate $U_{\ell+1}^i$, uniformly distributed on $[0, 1]$, independently of everything. If $U_{\ell+1}^i < p^i$, we add the neuron i in the set $\Gamma_{(\ell+1)\Delta}^0$ of spiking neurons.

- Construction of the cascade. Following Subsection 2.8, we let for $i \notin \Gamma_{(\ell+1)\Delta}^0$:

$$\hat{\bar{X}}_{1,(\ell+1)\Delta}^i := \hat{\bar{X}}_{(\ell+1)\Delta}^i + \sum_{j \in \Gamma_{(\ell+1)\Delta}^0} J_N^{j \rightarrow i},$$

and then $\Gamma_{(\ell+1)\Delta}^1 := \{i \in \{1, \dots, N\} \setminus \Gamma_{(\ell+1)\Delta}^0 : \hat{X}_{1,(\ell+1)\Delta}^i \geq X_F\}$. Iteratively, we define for any $i \in \{1, \dots, N\}$:

$$\hat{X}_{k+1,(\ell+1)\Delta}^i := \begin{cases} \hat{X}_{k,(\ell+1)\Delta}^i + \sum_{j \in \Gamma_{(\ell+1)\Delta}^k} J_N^{j \rightarrow i} & \text{if } i \notin \Gamma_{(\ell+1)\Delta}^k, \\ \hat{X}_{k,(\ell+1)\Delta}^i - \left[\frac{\hat{X}_{k,(\ell+1)\Delta}^i - X_R}{X_F - X_R} \right] (X_F - X_R) & \text{if } i \in \Gamma_{(\ell+1)\Delta}^k, \end{cases}$$

$$\Gamma_{(\ell+1)\Delta}^{k+1} := \left\{ i \in \{1, \dots, N\} \setminus \Gamma_{(\ell+1)\Delta}^k : \hat{X}_{k+1,(\ell+1)\Delta}^i \geq X_F \right\}.$$

– Final update. Call k_{\max} the first index when $\Gamma_{(\ell+1)\Delta}^{k_{\max}} = \emptyset$. And then we let

$$\bar{M}_{(\ell+1)\Delta}^i = \bar{M}_{\ell\Delta}^i + \sum_{k=0}^{k_{\max}-1} \mathbb{1}_{\Gamma_{(\ell+1)\Delta}^k}(i),$$

$$\bar{X}_{(\ell+1)\Delta}^i = \hat{X}_{k_{\max},(\ell+1)\Delta}^i.$$

Algorithm 1 Pseudo code for the simulation of a network of interacting neurons in case of hard threshold

Initialize the processes X^i and M^i of neuron i , for each $i = 1, \dots, N$;

repeat

 simulate the potential of each neuron for one time-step not considering interaction or resetting;

 add every neuron with potential above the threshold to the list of spiking neurons;

for i neuron still below the threshold **do**

 simulate the event “a Brownian bridge crosses the threshold” for neuron i ;

 update the list of spiking neurons;

repeat

 update the entire network by adding kicks caused by spiking;

 reset the spiking neurons;

 update the list of spiking neurons;

until the cascade of spiking neurons has been exhausted

until we have simulated the network for enough time-steps

3.3. Soft threshold. We now address a new model, which leads to a new algorithm. We explain below how this new model is connected with the previous, but, first, we want to make clear the reasons why we introduce this new model:

- One issue with the model developed previously is that it allows a cascade phenomenon; this turns out to be a hindrance from the numerical point of view. We are thus interested in having a smoother version.
- Anyway, the model studied up to now is not perfectly adapted to biological measurements: one main criticism (to which we already alluded in introduction) is precisely the fact that spikes are transmitted instantaneously to the post-synaptic neurons. Indeed, in (3), the post-synaptic neuron j receives a kick of size $J_N^{i \rightarrow j}$ exactly at the spiking time of the pre-synaptic neuron i . If its membrane potential exceeds the threshold, the neuron j also spikes at the same time.

The idea for solving this issue is to use a point process based model (see [5, 11] for related references), with the following main features: Neurons have a probability of spiking that is strictly less than one when their potential reaches the threshold (which explains why the threshold is no longer *hard*). Also the probability of having two spikes at the exact same time is zero (as long as the potential has a finite value) and so there is no more possibility of a cascade. A point process based system can be difficult to simulate, but there exist recipes for simplifying the creation of the algorithm. One of them is an algorithm developed by Ogata, see [21], specifically designed for simulating point processes. We make it clear below.

Our soft threshold model is a Markov process based upon the following general rule. The neuron i spikes at time t with rate $f^i(X_t^i)$, that is

$$\lim_{dt \rightarrow 0} \frac{1}{dt} \mathbb{P}(\text{neuron } i \text{ spikes in } [t, t + dt] \mid \sigma(X_s^i, s < t, i = 1, \dots, N)) = f^i(X_t^i).$$

Also, given the history before t , neurons behave independently until a new neuron spikes. In particular, for such a model, two neurons spike precisely at the same instant with probability 0, which explains why we have no more cascades. Between two spikes in the system, the membrane potentials of the neurons evolve independently according to the same diffusive dynamics as in (1); when a neuron spikes, all the membrane potentials receive a kick, as prescribed by the values of the synaptic weights.

Remark 11. (1) One can think of the hard threshold model as a particular case of the soft threshold model with the rate functions

$$f^i(x) = \begin{cases} +\infty & \text{if } x \geq X_F \\ 0 & \text{if } x < X_F, \end{cases}$$

in which case the rate functions are the same for all the neurons.

- (2) the choice $f^i(x) = C((x - X_F)_+)^p$ with a large $p \in \mathbb{N}$ and a good choice of constant C should be a good approximation of the hard threshold model. Notice that this intensity function is equal to zero for values of the potential that are below X_F , the latter still playing the role of a threshold: biological neurons indeed do not spike when their potential is below some threshold (actually it depends on the type of neurons we are dealing with, but this is mostly true for cortex neurons). When the potential exceeds X_F , the probability of spiking should quickly increase with the membrane potential so that the neuron will quickly spike.

The soft threshold model can be rigorously defined as a multivariate point process. Although we do not provide the definition explicitly, we borrow materials from this theory to construct our algorithm.

- First, we know that, given the membrane potentials X_t^1, \dots, X_t^N at a time t (t being typically an event of the multivariate point process), the first next event $t + T$ after t (or equivalently, the first time after t at which a new neuron spikes) is given by the first occurrence of a new point in a point process of intensity $(\sum_{i=1}^N f^i(X_{t+\theta}^i))_{\theta \geq 0}$.
- Second, given T , the label i of the spiking neuron is distributed proportionally to $f^i(X_{t+T}^i)$.

Although this construction is pretty simple, it is not possible to implement it directly in the form of a simulation method. So, in practice, we combine the last two points with a rejection procedure. Ideally, it reads as follows:

- For all, $i = 1, \dots, N$, we introduce a (predictable) process $(\tilde{f}_{t+\theta}^i)_{\theta \geq 0}$ such that $f^i(X_{t+\theta}^i) \leq \tilde{f}_{t+\theta}^i$, for all $\theta \geq 0$;
- provided we can do so, we simulate, conditional on the history up until time t , the next event (or next point) $t + T$ of a point process with intensity $(\sum_{i=1}^N \tilde{f}_{t+\theta}^i)_{\theta \geq 0}$;
- we choose the label i_0 of the spiking neuron proportionally to $\tilde{f}_{t+T}^{i_0}$;
- we simulate the membrane potential $X_{t+T}^{i_0}$ according to the sole diffusive dynamics prescribed before;
- we *accept* the spike of neuron i_0 at time $t + T$ with probability $f^i(X_{t+T}^{i_0})/\tilde{f}_{t+T}^{i_0}$. In this case, we update its membrane potential to its reset value $X_{t+T}^{i_0} = X_R$ and add the kick $J_N^{i_0 \rightarrow j}$ to the post-synaptic neurons j (i.e. such that $J_N^{i_0 \rightarrow j} \neq 0$);
- we then restart the procedure.

Of course, the choice of the “approximation functions” $(\tilde{f}_{t+\theta}^i)_{\theta \geq 0}$ is crucial, as the next event of the process $(\sum_{i=1}^N \tilde{f}_{t+\theta}^i)_{\theta \geq 0}$ should be easily simulated. In practice, functions $(\tilde{f}^i)_{i=1, \dots, N}$ are taken piecewise constant. We can even think of requiring the condition $\tilde{f}_{t+\theta}^i \geq f^i(X_{t+\theta}^i)$ for θ in a small neighborhood of t only and then of choosing $(\tilde{f}_{t+\theta}^i)_{\theta \geq 0}$ as a random function depending on the sole past of the system before t , in which case the next event $t + T$ is simulated as the next event in a standard Poisson process.

In fact, the condition $\tilde{f}_{t+\theta}^i \geq f^i(X_{t+\theta}^i)$ may be easily verified when we consider the deterministic part of the dynamics (with the sole drift b and without the Brownian motion), as the values of the process X^i are then easily controlled. However, because of the Brownian part in the dynamics, it is impossible to create a piecewise constant approximation function, as the support of a non-degenerate Gaussian random variable is the whole $(-\infty, +\infty)$. However the probability that these values be very large is quite small; as a result, we can choose the approximation function in such a way that $\tilde{f}_{t+\theta}^i \geq f^i(X_{t+\theta}^i)$ with very high probability for θ in the neighborhood of t .

For instance, if we consider that, between two spikes t and $t + T$, X^i has the following Ornstein-Uhlenbeck dynamics:

$$dX_\theta^i = -\lambda^i(X_\theta^i - a_i)d\theta + \sigma^i dW_\theta^i,$$

with X_t^i as initial condition and a_i as attractor, then we have the following explicit expression for the membrane potential:

$$X_\theta^i = a_i + \exp(-\lambda^i(\theta - t))(X_t^i - a_i) + \sigma^i \int_t^\theta \exp(-\lambda^i(\theta - s))dW_s^i, \quad \theta < t + T.$$

In words, the membrane potential follows Gaussian dynamics between two spikes and, using standard confidence intervals for the supremum of Gaussian processes, we can easily cook up the approximation function \tilde{f}^i .

Remark 12. (1) For very large N , we do not update \tilde{f}^i at each spiking times of one neuron. We only modify the value of \tilde{f}^{i_0} and \tilde{f}^j for neurons j s.t. $J_N^{i_0 \rightarrow j} \neq 0$.

- (2) The full network is implicitly updated after a fixed time interval: the time is segmented by a coarse time grid of scale Δ , and the \tilde{f}^i are computed within these intervals. At the end of one of these intervals, the maxima \tilde{f}^i are updated. We use this method in order to achieve a trade-off: if Δ is large then the maxima \tilde{f}^i are large which means high computational cost due to the percentage of rejected points whereas if Δ is small then the computational cost is high due to the frequent updates of the full network. The scale Δ is chosen arbitrarily from experience.

Summary. The algorithm can be found in a *pseudo-code form* in Algorithm 2 and is represented in Figure 4. It can be summarized in three steps: generate a spiking time t , select a neuron, apply a rejection sampling condition on the spike. This method is applied as many times as necessary for reaching a certain condition, typically until a final time is reached or a given number of spikes has been found. The rejection method is represented on step (4) in Figure 4, but several preliminary steps are necessary.

The step (0) in Figure 4 is to find a time interval that should contain the next event. This step is actually important for the speed of the algorithm. Events are exponentially separated, with the sum of the approximation functions as parameter, see step (1) in Figure 4:

$$T_{k+1} = T_k + \tilde{T}, \text{ with } \tilde{T} \sim \mathcal{E}(\sum_i \tilde{f}^i)$$

This step gives the time when the event is occurring. To find the neuron responsible for the event, we use again the sum of the approximation functions. The bigger the value of \tilde{f}_i , the higher the probability that the neuron i is the next one to spike.

Once we know which neuron spikes, we update its potential $X^i(T_{next})$ (step (3)). In this regard, it is important to remark that, while the event is categorized as T_{next} in step (1), it is regarded in the end as an event proper to neuron i (2 in the figure); we denote it by $T_{i,n}$.

The value of the potential X^i at time $T_{i,n}$ is needed to compute the value of the rate $f_i(X^i(T_{i,n}))$ and thus to classify the event between false and true spikes (step (4)). The value of a random variable uniformly distributed between 0 and \tilde{f}_i gives the classification: if it is less than $f(X^i(T_{i,n}))$, the spike is a true one; if it is greater, it is a false one. If the spike is a true one, the potential of the spiking neuron is reset to X_R while the potentials of all the postsynaptic (children) neurons are updated to the time of event T_{next} . The values of the synaptic weights $(J^{i \rightarrow j})_{j=1, \dots, N}$ are then added to their potentials (step (5)) (so that the rates $(f_j(X_{T_{i,n}}^j))_{j=1, \dots, N}$ may increase). If the spike was a false one, nothing happens.

The algorithms then loops back to step (0) until a given condition (number of spikes reached, time elapsed, etc.) is met.

Algorithm 2 Simulator

- 1: N : number of neurons
 - 2: Set $i = k = n = 0$ and $\tau_0 = T_0 = 0$
 - 3: Set $(T^j)_{j \in \{1, \dots, N\}} = (0)$ and $(X^j)_{j \in \{1, \dots, N\}} = (0)$
 - 4: Set $i = i + 1$ and generate τ_{i+1}
 - 5: Compute $S = \sum_{n=1}^N \tilde{f}_{\tau_i}^n$
 - 6: Set $\tilde{T} \sim \mathcal{E}(S)$ and $\Delta t = \Delta t + \tilde{T}$
 - 7: **if** $\tau_{i-1} + \tilde{T} > \tau_i$ **then**
 - 8: Set $i = i + 1$ and go to 4
 - 9: Set $k = k + 1$ and put $T_k = T_{k-1} + \tilde{T}$ ▷ System spiking time
 - 10: Put $u \sim \mathcal{U}([0, 1])$
 - 11: $n \leftarrow \text{argmin}_{n \in \{1, \dots, N\}} \left(\sum_{j=1}^N \tilde{f}_{\tau_i}^j * u < \sum_{j=1}^n \tilde{f}_{\tau_i}^j \right)$
 - 12: Put $t = T_k - T^n$ and $T^n = T_k$
 - 13: Put $X^n = a^n + e^{-\lambda^n t} (X^n - a^n) + \sigma^n \mathcal{N}(0, (1 - e^{-2\lambda^n t}) / (2\lambda^n))$
 - 14: Put $u \sim \mathcal{U}([0, 1])$
 - 15: **if** $\tilde{f}_{\tau_i}^n * u < f(X^n)$ **then**
 - 16: **for all** j **do** 1N
 - 17: **if** $j \neq n$ **then**
 - 18: $X^j = a^j + e^{-\lambda^j t} (X^j - a^j) + \sigma^j \mathcal{N}(0, (1 - e^{-2\lambda^j t}) / (2\lambda^j))$
 - 19: $X^j = X^j + J^{n \rightarrow j}$
 - 20: Put $\tau_i = T_k$ and go to 4
-

The algorithm (for Ornstein-Uhlenbeck subthreshold dynamics).

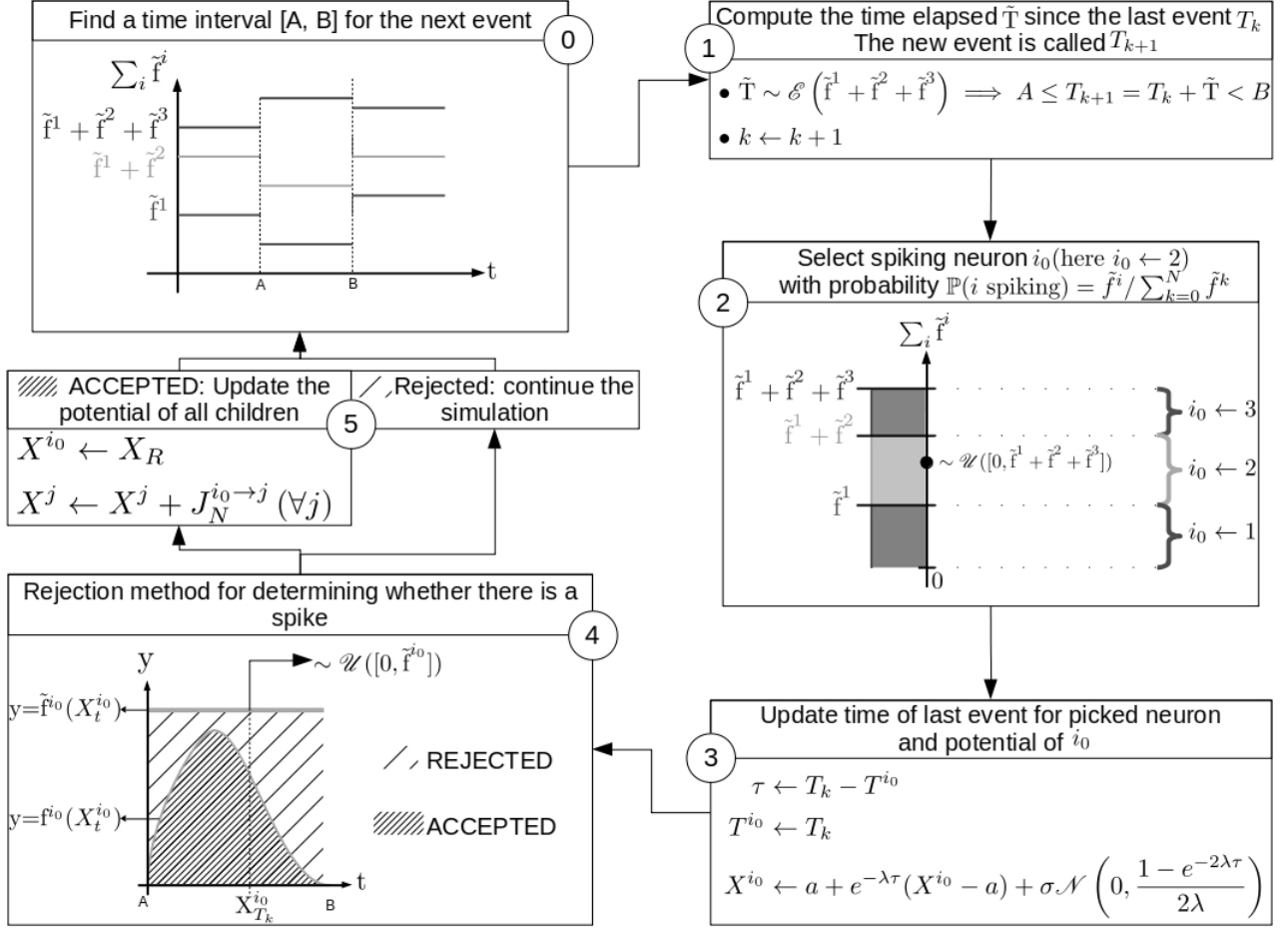


FIGURE 4. Illustration of Algorithm 2 for a system composed of 3 neurons.

The steps (0)→(1)→(2)→(3)→(4)→ (5) are described in the text.

Comparison between clock-driven and event-driven algorithms. We now argue why the event-driven algorithm may be more advantageous than the clock-driven one.

We first recall that the clock-driven method is based upon a time discretization. The state of the system is indeed approximated at discrete times of the form $(T_0 + \ell\Delta)_{\ell \in \mathbb{N}}$, for a given time step Δ . The advantage is pretty clear: The clock-driven method may be easier to implement than the event-driven one. Still, it has a major drawback: the time mesh disregards the own clock of the system; the former is indeed uniform whilst the latter is deeply heterogeneous since spikes may occur much more frequently at certain periods. In other words, the time mesh chosen in the clock-driven method may not be adapted to the dynamics of the system (we did not explore adaptive time meshes since the event-driven method is automatically adapted). As a result, it may be harder for the clock-driven algorithm to capture the right behavior, leading to precision, stability and complexity issues.

For instance, precision issues may arise because spikes (at least in the simplest form of the clock-driven method) can only occur at the nodes of the time mesh. For sure, this creates a local small error, that may accumulate on the long run. Whilst sophisticated strategies may be implemented to mitigate this effect (for instance the dynamics between two consecutive nodes may be reconstructed *a posteriori* in the form of a bridge), a common (and simple) solution is to take a very small time step Δ . However, the smaller the step Δ the higher the computational cost.

For sure, choosing Δ small has a low interest when the activity of the network is low: When spikes are rare, the computational effort that is required for handling a small Δ looks somewhat disproportionate.

Conversely, Δ has to be chosen very small in order to account for blow-ups. As stated above, this increases the complexity of the method.

These basic observations explain the need for an alternative, event-driven, method. By focusing on the spikes generation (and only computing that), the event-driven approach may indeed reduce the global complexity. Also, times at which the dynamics are simulated are no longer multiples of Δ but can now potentially take any decimal value within the precision range of the machine. The values are then easily precise up to 10^{-15} on modern machines.

3.4. Graph management. The storage of the interaction matrix J_N is a strong limitation in our algorithm. To simulate a network with N neurons, we need to know the values of N^2 synaptic weights $J_N^{i \rightarrow j}$. In practice, the size of the memory requested to store this matrix is one of the main limitation in our algorithm. For instance, in the classical algorithm used for creating a N -node Erdős-Renyi directed graph, a matrix of size N^2 is created, where each element indicates whether there is a connection between two neurons; typically, this may be an issue when N is greater than 10^6 . Indeed, it is impossible to store on a laptop a graph with one million of nodes, or a network with one million of neurons.

The main idea we propose here is to slightly change the classical method for generating the synaptic weights (when given as the realizations of independent and identically distributed variables as it is the case in the Erdős-Renyi graph); in a sentence, our strategy is: “instead of storing N rows of size N , store only N numbers, namely store for each row one number that encodes the whole row”.

We call our new algorithm a *reconstruction* algorithm. Let us rephrase the main idea: instead of storing the full matrix of interaction, we store N values, say $S_0, S_N \dots, S_{N(N-1)}$ (the fact the indices are in the form Ni is explained in Figure 5 below). The knowledge of S_{Ni} is sufficient to compute the row $J_N^{i \rightarrow \bullet}$ of the synaptic weights from i to all the other neurons. In other words, instead of requesting a memory stack of order $N \times N$, we just request to a smaller memory stack of order N . Of course, the price to pay for this operation is that, each time we need the elements of the row number i , we have to recompute from scratch the whole vector of interactions pointing from i . As we make clear below, this method is numerically efficient if, first, the matrix is static (i.e., the values do not evolve in time), and second, we do not use it too often; in fact, both conditions are fulfilled in our setting.

The whole trick is based on the fact that, from the numerical point of view, the matrix J_N is obtained by a simulation technique, which is in fact deterministic! Consider indeed a Pseudo Random Number Generator (PRNG) in a state S_0 . If one uses this PRNG once, a random number r_0 is produced and the state of the PRNG is changed from S_0 to S_1 . After N iterations, the PRNG is in a state S_N and N random numbers have been returned. Now if we put the PRNG back to state S_0 , and ask again for a number, then r_0 will be returned again and the RNG will be back in state S_1 ; after N iterations, the PRNG is again in state S_N and the N random numbers that have been returned are the same as the N numbers that had been initially generated. So, going back to the matrix J_N , we can store the states S_{Ni} of the PRNG at the beginning of the simulation of the row $i + 1$. Each time we need the values of the weights $J_N^{i_0 \rightarrow j}$, we just change the state of the PRNG to S_{Ni_0} and generate again (with the PRNG) the *pseudo-random* values $J_N^{i_0 \rightarrow j}$. Obviously, putting the PRNG in the same state gives the same *pseudo-random* result. We make this clear in Figure 5 below.

This method is easily implemented in the case of an Erdős-Renyi graph (which is our benchmark example), see also Algorithm 3. Instead of storing the connections in a matrix, one stores only the first states $(S_{Ni})_{i=0, \dots, N-1}$; at the end of the day, the memory print is much lower... but the computational cost is higher. To make it clear, this algorithm is not very efficient if one needs to often access the *individual* entries of the matrix, or if one wants a dynamical system where the graph connections are allowed to evolve in time. But in our case the connections are fixed once for all at the beginning of the simulation and the only moment we need them is to update the potential of the children after a spike; in the latter case (and this the key point), what we want is exactly the full row of the matrix.

$$\begin{array}{lcl}
 \text{Classical method :} & & \begin{array}{l} S_0 \rightarrow \left(\begin{array}{cccc} J_{11} & J_{12} & \cdots & J_{1N} \\ J_{21} & & & J_{2N} \\ \vdots & & & \vdots \\ J_{N1} & J_{N2} & \cdots & J_{NN} \end{array} \right) = J \\ S_N \rightarrow \left(\begin{array}{cccc} J_{11} & J_{12} & \cdots & J_{1N} \\ J_{21} & & & J_{2N} \\ \vdots & & & \vdots \\ J_{N1} & J_{N2} & \cdots & J_{NN} \end{array} \right) = J \\ \vdots & & \vdots \\ S_{N(N-1)} \rightarrow \left(\begin{array}{cccc} J_{11} & J_{12} & \cdots & J_{1N} \\ J_{21} & & & J_{2N} \\ \vdots & & & \vdots \\ J_{N1} & J_{N2} & \cdots & J_{NN} \end{array} \right) = J \end{array} \\
 \\
 \text{Reconstruction method :} & & J = \left(\begin{array}{c} S_0 \\ S_N \\ \vdots \\ S_{N(N-1)} \end{array} \right) \rightarrow \begin{array}{cccc} J_{11} & J_{12} & \cdots & J_{1N} \\ J_{21} & & & J_{2N} \\ \vdots & \vdots & & \vdots \\ J_{N1} & J_{N2} & \cdots & J_{NN} \end{array}
 \end{array}$$

FIGURE 5. The two graph management methods summarized here. The \rightarrow means “generate the row on the right”, the $(S_i)_{i \in \{0, \dots, N(N-1)\}}$ are the state of the PRNG before the generation of the row and (\cdot) is what is kept in memory (graph in the classical method, vector of PRNG’s states in the reconstruction method)

Algorithm 3 Generation of a vector of PRNG states

```

1: PRNG: Pseudo Random Number Generator
2: p: probability of connection between two neurons
3: N: number of neurons in the system
4:  $(V_i)_{i \in \{1, \dots, N\}}$ : vector of PRNG status
5: for i do 0N-1
6:    $V_i \leftarrow S_{i*N}$ 
7:   for j do 1N
8:     PRNG
    
```

▷ State change from $S_{i*N+j-1}$ to S_{i*N+j}

Algorithms.

Remark 13. We have to use another random generator number for the simulations of the Brownian motions and the Poisson Processes.

Algorithm 4 gives a comparison between the two (classical and reconstruction) methods:

Algorithm 4 Comparison of usage between classical method and reconstruction

$(M_{ij})_{(i,j) \in \{1, \dots, N\}^2}$: Interaction matrix

Neuron i is spiking

▷ Using matrix graph

for j **do** 1N

if $\mathcal{M}_{i,j} = 1$ **then**

 Update potential of neuron j

▷ Using reconstruction

RNG.SETSTATE(V_i)

for j **do** 1N

if BERNOULLI(RNG)=1 **then**

 Update potential of neuron j

A natural question is the question of the complexity of such a reconstruction-based method compared to the complexity of the classical one. This is a broader matter that is addressed in the next section.

3.5. Complexity: memory and instructions. We remind the user that algorithmic complexity must be read as a function relating the input of an algorithm (more precisely its size) and the number of steps it takes (its time complexity) or the number of storage locations it uses (its space complexity). The asymptotic values of this function (for larger and larger sizes of the input) are noted with a big \mathcal{O} notation.

Of course, the reader must remember that two algorithms with the same order of complexity (or two implementations of the same algorithm) may execute with different amounts of resources because of a multiplicative factor between the two (which will become somehow insignificant asymptotically, but which is not for inputs of

small size). Having in mind this limitation, the analysis we provide below focuses on the comparison between the orders of complexity.

| | Time complexity at creation | Time complexity during usage | Space complexity |
|-----------------------|-----------------------------|------------------------------|--------------------|
| Reconstruction method | $\mathcal{O}(N^2)$ | $\mathcal{O}(N)$ | $\mathcal{O}(N)$ |
| Interaction matrix | $\mathcal{O}(N^2)$ | $\mathcal{O}(1)$ | $\mathcal{O}(N^2)$ |

TABLE 1. Number of operations to create the matrix (col. 1); Number of operations at each call (col. 2); Memory complexity (col. 3).

It must be stressed that the only input that we regard in the analysis of the complexity is the number N of neurons. In particular, all the routines that do not depend explicitly on N do not really appear in the complexity, whilst they could have a non-negligible cost. For instance, PRNG takes time, but it does not show up in the final computation of the complexity.

The first complexity column is given as an example of how the complexity is computed. In the classical algorithm for generating an interaction matrix, for all the parents, we must look at all the other neurons in order to determine whether they are children of the parent. Hence the double loop, leading to N^2 . In the case of the reconstruction method presented above, the number of children of a given parent is determined before hand, and in the second loop their index.

It must be also stressed that, in practice, the time complexity at creation of the reconstruction method is lower than the complexity of the interaction method; theoretically, this is not the case as one should take into account the worst case scenario (i.e, the complete graph).

As for the last two columns, the space complexity is obviously smaller for the reconstruction method, making it very relevant for big graphs. Still, the time complexity using the vector of PRNG states is worse than the time complexity of the interaction method, but, as we already commented, this is not such a hindrance in our case.

Acknowledgment. The authors thank Structuring program UCA “Cognitive systems, normality and pathology of the human brain, computational neuroscience”, the project “Modélisation Théorique et Computationnelle en Neurosciences et Sciences Cognitives” from University Côte d’Azur, Université Nice Sophia Antipolis, Centre Hospitalier Universitaire de Nice, CNRS and INRIA, for the financial support. This work has been supported by the French government, through the UCAJEDI Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-15-IDEX-01. ”

REFERENCES

- [1] María J. Cáceres, José A. Carrillo, and Benoît Perthame. Analysis of nonlinear noisy integrate & fire neuron models: blow-up and steady states. *J. Math. Neurosci.*, 1(1):7, Jul 2011.
- [2] José A. Carrillo, María D. M. González, Maria P. Gualdani, and Maria E. Schonbek. Classical solutions for a nonlinear Fokker-Planck equation arising in computational neuroscience. *Comm. Partial Differential Equations*, 38(3):385–409, 2013.
- [3] José Antonio Carrillo, Benoît Perthame, Delphine Salort, and Didier Smets. Qualitative properties of solutions for the noisy integrate and fire model in computational neuroscience. *Nonlinearity*, 28(9):3365–3388, 2015.
- [4] Eleonora Catsigeras and Pierre Guiraud. Integrate and fire neural networks, piecewise contractive maps and limit cycles. *J. Math. Biol.*, 67(3):609–655, 2013.
- [5] Anna De Masi, Antonio Galves, Eva Löcherbach, and Enrico Presutti. Hydrodynamic limit for interacting neurons. *J. Stat. Phys.*, 158(4):866–902, 2015.
- [6] François Delarue, James Inglis, Sylvain Rubenthaler, and Etienne Tanré. Global solvability of a networked integrate-and-fire model of McKean-Vlasov type. *Ann. Appl. Probab.*, 25(4):2096–2133, 2015.
- [7] François Delarue, James Inglis, Sylvain Rubenthaler, and Etienne Tanré. Particle systems with a singular mean-field self-excitation. Application to neuronal networks. *Stochastic Process. Appl.*, 125(6):2451–2492, 2015.
- [8] Sylvain Delattre, Giambattista Giacomini, and Eric Luçon. A note on dynamical models on random graphs and Fokker-Planck equations. *J. Stat. Phys.*, 165(4):785–798, 2016.
- [9] Paul Erdős and Alfred Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.
- [10] Paul Erdős and Alfred Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61, 1960.
- [11] Nicolas Fournier and Eva Löcherbach. On a toy model of interacting neurons. *Ann. Inst. Henri Poincaré Probab. Stat.*, 52(4):1844–1876, 2016.
- [12] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical-reactions. 81:2340–2361, 12 1977.
- [13] Emmanuel Gobet. Weak approximation of killed diffusion using Euler schemes. *Stochastic Process. Appl.*, 87(2):167–197, 2000.
- [14] Alasdair J. Graham and David A. Pike. A note on thresholds and connectivity in random directed graphs. *Atl. Electron. J. Math.*, 3(1):1–5, 2008.
- [15] Ben Hambly, Sean Ledger, and Andreas Søjmark. A McKean–Vlasov equation with positive feedback and blow-ups. *ArXiv e-prints*, 2018.
- [16] Alan L. Hodgkin and Andrew F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117(4):500–544, 1952.
- [17] Louis Lapicque. Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol. Gen.*, 9:620–635, 1907.
- [18] Timothy J. Lewis and John Rinzel. Dynamics of spiking neurons connected by both inhibitory and electrical coupling. *J. Comput. Neurosci.*, 14(3):283–309, 2003.

- [19] Sergey Nadtochiy and Mykhaylo Shkolnikov. Particle systems with singular interaction through hitting times: application in systemic risk modeling. *ArXiv e-prints*, 2017.
- [20] Sergey Nadtochiy and Mykhaylo Shkolnikov. Mean field systems on networks, with singular interaction through hitting times. *ArXiv e-prints*, 2018.
- [21] Yoshihiko Ogata. On Lewis' simulation method for point processes. *IEEE Trans. Inform. Theory*, 27:23–31, 1981.
- [22] Srdjan Ostojic, Nicolas Brunel, and Vincent Hakim. Synchronization properties of networks of electrically coupled neurons in the presence of noise and heterogeneities. *J. Comput. Neurosci.*, 26(3):369–392, 2009.
- [23] Ilona Palásti. On the strong connectedness of directed random graphs. *Studia Sci. Math. Hungar.*, 1:205–214, 1966.
- [24] Alain-Sol Sznitman. Topics in propagation of chaos. In *École d'Été de Probabilités de Saint-Flour XIX—1989*, volume 1464 of *Lecture Notes in Math.*, pages 165–251. Springer, Berlin, 1991.

MATHEMATICS INSTITUTE, UNIVERSITY OF WARWICK, COVENTRY, UNITED KINGDOM.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF PISA, PISA, ITALY.

UNIVERSITÉ CÔTE D'AZUR, CNRS, I3S, FRANCE

LABORATOIRE JEAN KUNTZMANN, UNIVERSITÉ GRENoble ALPES (UFR IM2AG), GRENoble, FRANCE

UNIVERSITÉ CÔTE D'AZUR, CNRS, LJAD, FRANCE

UNIVERSITÉ CÔTE D'AZUR, INRIA, FRANCE